今回の USB メモリの中身は picture ディレクトリと .st が4つ

ディレクトリ構成 pictures Smalltalk.app VisualWorksWithJun/

File Browser を開いて SSK_PaneMVC_20110706.st を File in



inspect して、ワークスペースを開いて



self alignmentSymbol: #topRight. self model changed.

と入力して、Do itしてみる 右寄りになっている



下記のように入力してみると、 self isScaling: false. self model changed.



画像のサイズが Window に合わせて変更されなくなる

下記のように入力してみると self keepAspect: false. self model changed.



アスペクト比が保持されず、 View いっぱいに表示されるようになる

Q. Model が変わってないのに、 model changed って変じゃないの? A. Object の changed を参照

Object の changed を参照 update が呼ばれるのだが、これもやはり Object に存在する 追いかけていくと、 invalidate というものが呼ばれている

この invalidate は呼ばれるとすぐに動き出すという保証はない 他にすることがないので、すぐ動いているように見えるだけである すぐに呼ぶ必要がある場合は invalidateNow というものを実行すれば良い

デフォルトが invalidate な理由は、再描画の際に、手間を減らすため

Implementors of Selector で中身を見てみる



見比べてみると、

invalidate

"Invalidate the receiver's bounding box.

Propagate the damage rectangle up the containment hierarchy. This will result in a displayOn: aGraphicsContext being sent to the receiver."

container == nil

ifFalse: [self invalidateRectangle: self bounds]

invalidateNow

Invalidate the receiver's bounding box. Propagate the damage rectangle up the containment hierarchy. This will result in a displayOn: aGraphicsContext being sent to the receiver."

container == nil ifFalse: [self invalidateRectangle: self bounds repairNow: true]

この部分が違うだけである

複数の Window が重なっている場合に、 invalidate だと、重なっている部分は再描画されないのだが、 invalidateNow は全ての Window を書き直すので、負荷が大きくなる

-	-	-	-	-	-			
		-	-	-	-	-	-	

赤色部分が無駄に描画されることになる ただし、 デバッグをしているときは、 invalidateNow を用いないと、いつまで経っても Window の変化が起きないという事になるかもしれないので、使ってみると良い

=====復習終わり

=====ちょっと脱線

コードの読み方 何の動きか分からないところをコメントアウトしてみたり、分岐するところをあえて潰してどのように振る舞うかを試してみたり 例えば、 displayOn: graphicsContext "self model picture ifNotNil: [anImage 1 I messageSelector scaledImage aRectangle 1 messageSelector := self alignmentSymbol. scaledImage := self isScaling ifFalse: [anImage shrunkenBy: self scalingFactor reciprocal] ifFalse: [anImage yourself]. aRectangle := scaledImage bounds. aRectangle i= scaledImage bounds. aRectangle i= scaledImage bounds. aRectangle i= scaledImage bounds. aRectangle i= scaledImage bounds. scaledImage displayOn: graphicsContext at: aRectangle origin]." self model label ifNotNil:

ifNotNil: [:aString I I aComposedText aPoint I aComposedText := aString asComposedText. aPoint := 4 @ 0. graphicsContext paint: ColorValue white. (-1 to: 1) do: [:y I (-1 to: 1) do: [:x I aComposedText displayOn: graphicsContext at: aPoint + (x @ y)]]. graphicsContext paint: ColorValue black. aComposedText displayOn: graphicsContext at: aPoint]

0 0	Pane Example
BernhardRiemann	GeorgeBoole

そういう役割のコードか!と確認できる

0 0		Debug It		
Stack Method Ed	it Execute Correct	t		
1. C 🐨 🛈		→]← 🌢 📗	×	
PaneView>>unbound PaneView(Object)> PaneView(Object)> optimized [] in F BlockClosure>>on: optimized [] in F	Method >performMethod:a 'aragraphEditor>> do: 'rocess class>>fo	rguments: debug: rBlock:priorit	al: false r: a PaneView	
self keepAspect: self model chang	false. ed.			
self container eventHandlers properties model controller alignmentSymbo isScaling				<i>ti</i> s

self halt. でプレイクポイントを入れて、デバッガを起動させることが出来る デバッガから抜け出せば普通に動き出すことが出来る

Debug it すると、それを実行する直前で止まって、デバッガが動き出す

これを使えば、実行しながら、何処がどう動いているのか一歩ずつ確実に追いかけることが出来るのである!

=====脱線終わり

黄色ボタン(いわゆる右クリック)のメニューに inspect だけではなく、 topRight や isScaling を false にしたり出来るようにしたい Menu Editor を起動

0	۱ 🔘 (VisualWorl	ks NonCo	mmercial /Users/	fumiya/Doo	cuments/src/S]
File	Systen	n Browse	Debug	Painter Store Too	ls Window	Help	
∛ ¦a)	X 😥	°M °S	💉 New Canvas	d a		
Paneo Paneo Paneo Paneo Paneo Paneo	Contro Contro View>> View>> View>> View>>	oller>>red bler>>red bounds: 0 bounds: 0 bounds: 0 bounds: 0 bounds: 0	ButtonP ButtonP 0 @ 0 co 0 @ 0 co 0 @ 0 cor 0 @ 0 cor	Menu Editor Image Editor Hot Regions I Resource Fin ner: 283 @ 404 19 ner: 283 @ 404 19	Editor der 9:38:22 9:38:22	ļ	
New Ite	muc			Menu Editor			
Menu	Edit	Move Vie	ew Test	Menu Parto			
~	-2	New Item	Χħΰ	】× 名切.			
Inspe		New Sepa New Subr	arator nenu Item	Basic Details	State		
	X	Cut Item		₩Shift+Ctrl+X	Inspect		
	Ъ	Copy Item		₩Shift+Ctrl+C			
	Ĥ	Paste Item	n	₩Shift+Ctrl+V			
	\times	Delete Iter	n	₩Shift+Ctrl+D			
				Value:			
				ID:			
_							2
	0) 🔿		Menu	Leditor		
	Menu	Edit Mo	ve View	Test			

Menu Ed	it Me	ove	View	Tes	t					 	
× =	-		X	ĥ	Û	×	+1	*8		 	
V 1			<i>.</i>				Aasic abel: .abel: .ookup Catalog /alue: D:	Details key:	State Inspect #inspect I	 	
					_	_					

Install する

00) 😁			Menu Editor	
Menu	Edit Move New Load	View Tes %Alt+N %Ctrl+O	st	< 1 13 1	
- V	install	₩Alt+C		Basic Details	State
Inspe	Read Apply Close	₩Alt+Q		Label: String: Lookup key: Catalog: Value: ID:	Inspect #inspect
			_		



PaneExample の Class の resources に yellowButtonMenu が増えている

\varTheta 🔿 🔿 PaneExample class>	>yellowButtonMenu
Browser Edit Find View Package Class Protocol Method T	Tools Help
🔁 🔭 🔍 🕂 🐂 🎾 🎢 🎽 🕥 🍽 Find:	
Package Class	Instance Class Shared Variable Instance Variable
Base VisualWorks • PaneController	defaults yellowButtonMenu
Glorp PaneExample	examples
PaneWodel	resources
Les Pane	
E StoreBase	
Tools-IDE	
🚳 Assets	
Visual Source Comment Definition Rewrite Code Critic	
Edit Menu	÷
Method: #yellowButtonMenu (resources)	Package: Pane

PaneController σ Instance σ events σ yellowButtonPressedEvent: event $\epsilon \wedge \lambda z u$

yellowButtonPressedEvent: event "マウスの右ボタンが押されたときの処理をする。"

I aMenu aSymbol I aMenu := PaneExample yellowButtonMenu. aSymbol := aMenu startUp. (aSymbol isKindOf: Symbol) ifTrue: [self view perform: aSymbol]. この様に変更する

0 0				Menu Editor
Menu Edit	Move	View	Test	
V 🗄 🕯	3 4	Å	h (1	× ₁∃ ⊒ ⊒ ⊒.
scaling				Basic Details State
Inspect	_			Label:
				String: scaling
				Lookup key:
				Catalog:
				Value: #scalingOnOff
				ID:

scaling を追加する



今回は別の View に同じメニューが存在してもおかしくないのだが、場合によっては、異なる View は異なるメニューが欲しいかもしれない だから、先ほどの物ではダメなので、書き換え

yellowButtonPressedEvent: event "マウスの右ボタンが押されたときの処理をする。"

I aMenu aSymbol I aMenu := self model yellowButtonMenu. aSymbol := aMenu startUp. (aSymbol isKindOf: Symbol) ifTrue: [self model perform: aSymbol]. ^nil

menu accessing を追加

^nil

Panemodel>menu accessing
ew Package Class Protocol Method Tools Help
* 🕅 * 🖕 * 🎦 🐐 M Find:
Instance Class Shared Variable Instance Variable
s A PaneController 4 accessing
🚺 📰 PaneExample 🚺 initialize-release
PaneModel 🕕 menu accessing
• PaneView
Definition Rewrite Code Critic
i argument names
g purpose of message"
lable names
nenu accessing に 追加
ellowButtonMenu
^PaneExample yellowButtonMenu
nenu messages に追加 callingOnOff
I aView aBoolean I aView := self dependents first. aBoolean := aView isScaling. aView isScaling: aBoolean not. self changed
′iew が alignmentSymbol isScaling keepAspect を持っているのは実は変なのではないだろうかという事に nenu accessing の yellowButtonMenu も誰が処理をするのか分からない状態
iou が olignmentCumbel ioCooling loop Appent の桂根を持っているから Niou 気に囲軟がさいているが Medal がる

View が alignmentSymbol isScaling keepAspect の情報を持っているから、 View 毎に調整がきいているが、 Model がこれらの値を持ち出すと、 View 毎に調整がきかないように なってしまう

さて、どうしましょう? 解決方法 yellowButtonPressedEvent: event "マウスの右ボタンが押されたときの処理をする。"

I aMenu aSymbol I aMenu := self model yellowButtonMenu.

alvenu = self model yellowButtonMenu. aSymbol := aMenu startUp. (aSymbol isKindOf: Symbol) ifTrue: [self model perform: aSymbol]. ^nil

多く見かける物は、赤色部分の実装が View が処理をするのか、 Model が処理をするのか判別するように書いてある



例えば、エディタでメニューを出してみると、処理すべき物によってセパレータで分かれているような状態 メニューを出すときはいくつか存在する物を合体させて表示するような感じ

実際に動かしてみながら、どうするか考えてみよう

PaneModel に追加する (Class resources に入る)

yellowButtonMenu "Tools.MenuEditor new openOnClass: self andSelector: #yellowButtonMenu" <resource: #menu> ^#(#{UI.Menu} #(#(#{UI.Menultem}) #rawLabel: 'Select image file' #value: #selectImageFile) #(#{UI.Menultem} #rawLabel: 'Inspect' #value: #inspect)) #(1 1) nil) decodeAsLiteralArra

 $\mathsf{PaneModel}\ \mathcal{O}\ \mathsf{Instance}\ \mathcal{O}\ \mathsf{menu}\ \mathsf{messages}\ \mathcal{O}\ \mathsf{scalingOnOff}\ \mathfrak{E}\ \mathsf{remove}$

PaneModel Instance menu accessing $\ensuremath{\mathcal{O}}$ yellowButtonMenu

^self class yellowButtonMenu

paneView に書き加える paneView1

> I aModel aView I aModel := self createPaneModel: 'BernhardRiemann.jpg'. aView := PaneView model: aModel. aView alignmentSymbol: #topRight. ^aView

paneView2

I aModel aView I aModel := self createPaneModel: 'GeorgeBoole.jpg'. aView := PaneView model: aModel. aView alignmentSymbol: #bottomLeft. ^aView alignmentSymbol: #bottomLeft.

この用に alignment は View に表示される時点では、既に決まっている物なのではないだろうか わざわざ、ユーザが変更できる必要ってある? (プログラマーが描画位置は全て決定しているのではないかい?)

(誰が処理をするのか区別を付ける方法) Q.(案1)

メニューの階層化をして、それぞれ誰が処理するかを判別できるように出来ないか

A. Menu のサブクラスを作って、機能拡張すれば出来るが、デフォルトでは出来ない 結構こういうことをしている人はいる

Ackage Class	Set as Current Default Namespace	Instance Class Shar
- interf	Store rei Browse	nt 🚺
- interi	Find Package Find Namespace	sit
- Interf	Ma Find Class	
- 🏐 Interf - 🏐 Interf	Find Shared Variable Sort	
Comment Prop	File Into File Out	ode Critic

ontains implementation classes for various user actions that need to be tra

(案2)

メニューーつーつにタグを付けて、それで処理すべき MVC のいずれかを判断するようにしよう

0 0	Menu Editor
Menu Edit Move View Test	
V 222 × b	à× +∃ ;
Select image file	Basic Details State
Inspect	Label:
	String: Select image file
	Lookup key:
	Catalog:
	Value: #selectImageFile
	ID: #view

ID の部分に #view などと追加する

と... この先、この実装をしようとしたところで時間切れ (正確にはもう少し進んだが、メモを取りきれず…)