

USB メモリの中身

picture

.st ファイル

VisalWorks

RBCodeHighlighting_MoveEverythingInThisFolderToContributed

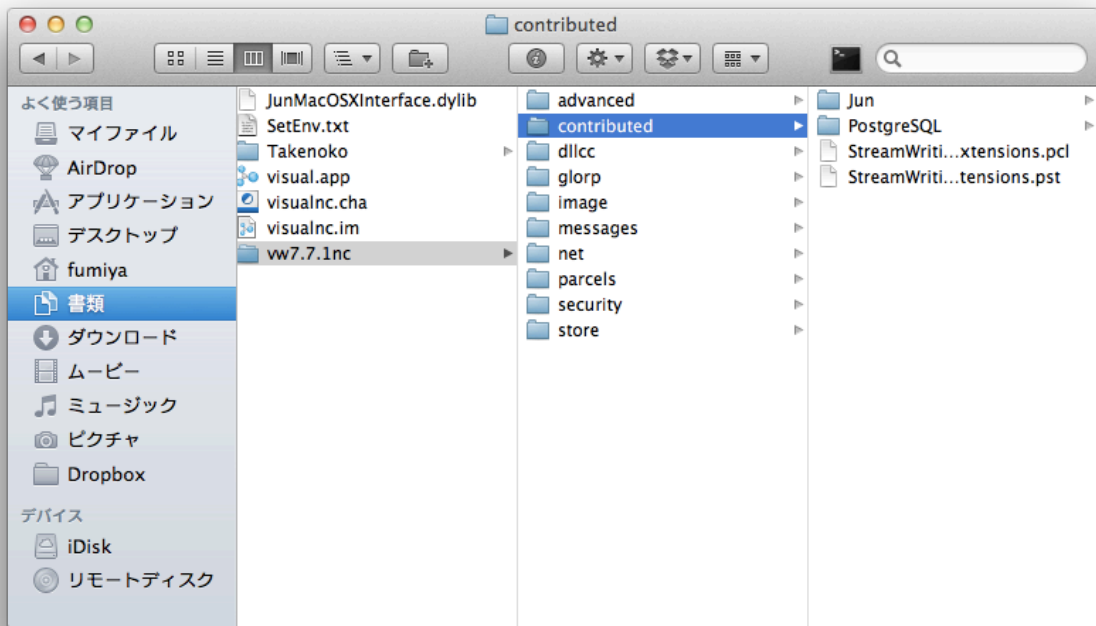
RBCodeHighlighting_MoveEverythingInThisFolderToContributed とは
というものがある

ソースコードをカラーリングしてくれるもの

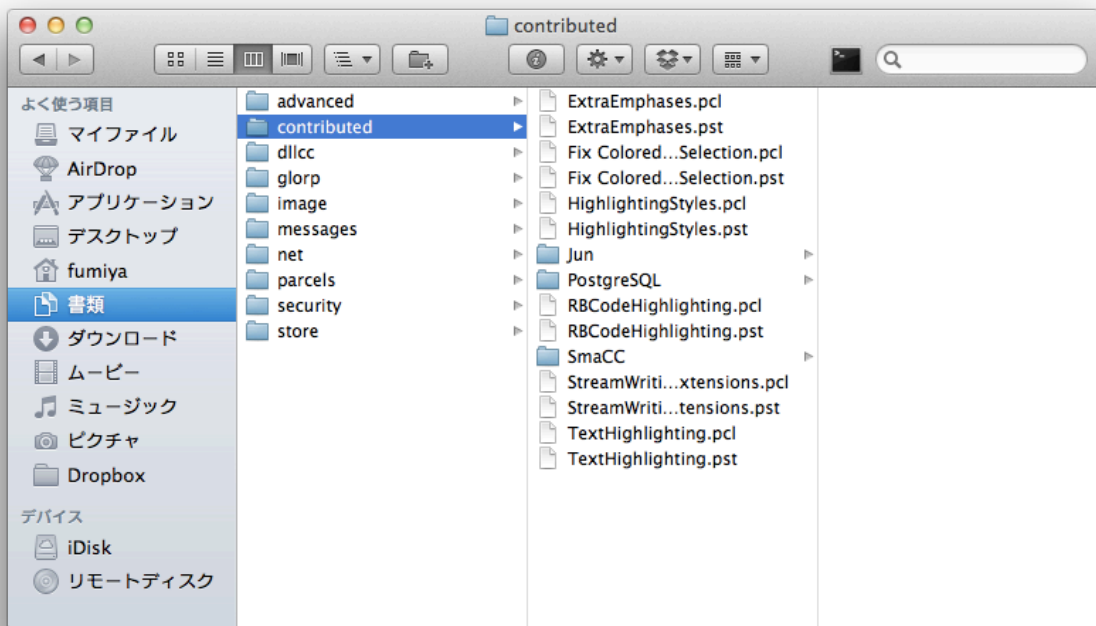
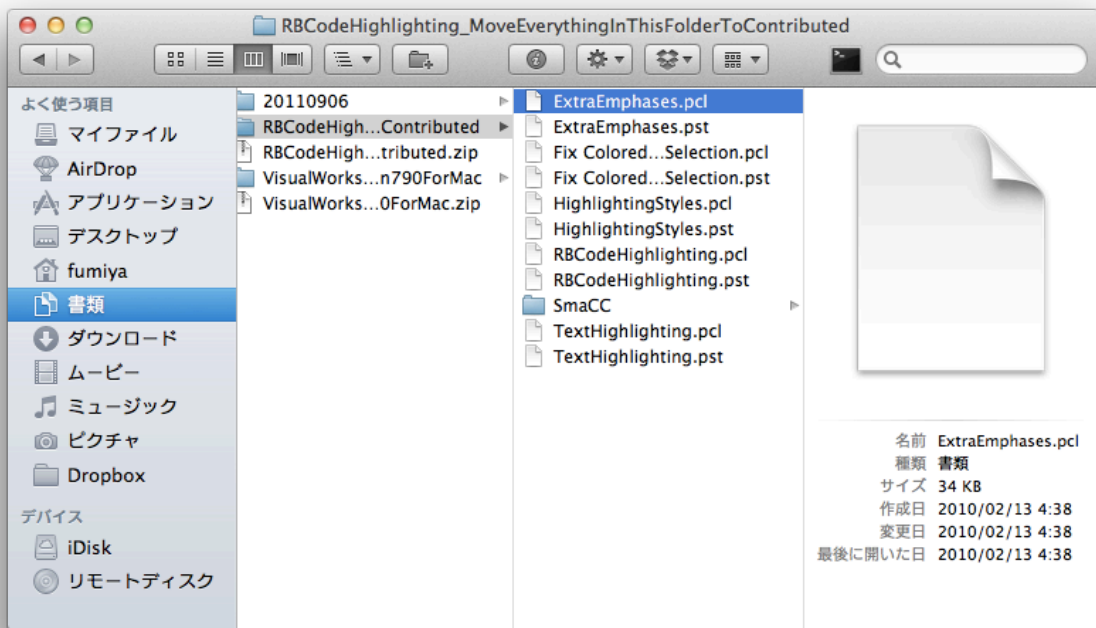
RB とはリファクタリングブラウザのこと

使い方

ファイル名に書いてあるとおり、すべてのものを Contributed へ移す

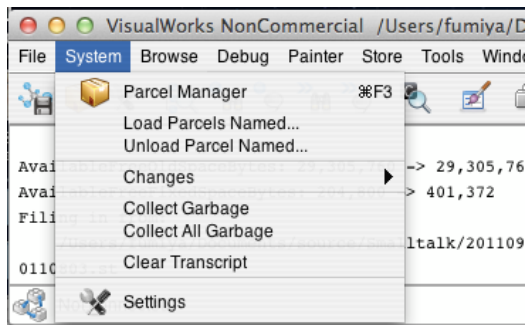


Contributed は Smalltalk に寄贈されたものたちが入っているらしい



こんな感じ

設置しただけでは動かないので読み込む

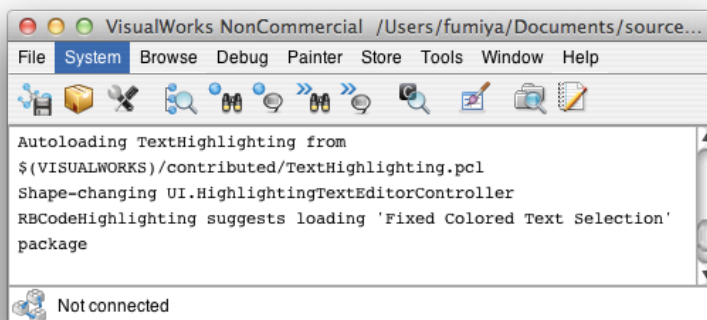


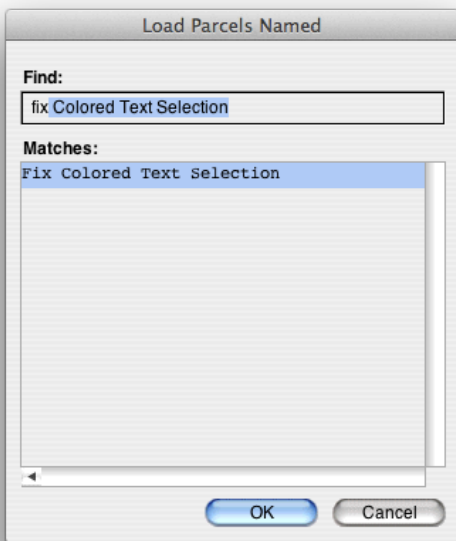
Load Parcels Named を選択



RBCodeHighlighting
を選択

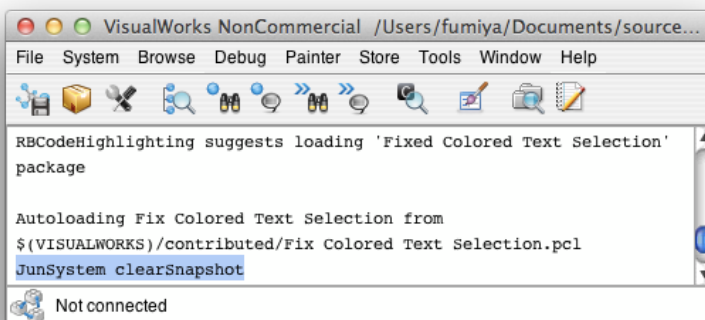
Transcript には Fixed うんたらと書かれているが、Fix うんたらの typo



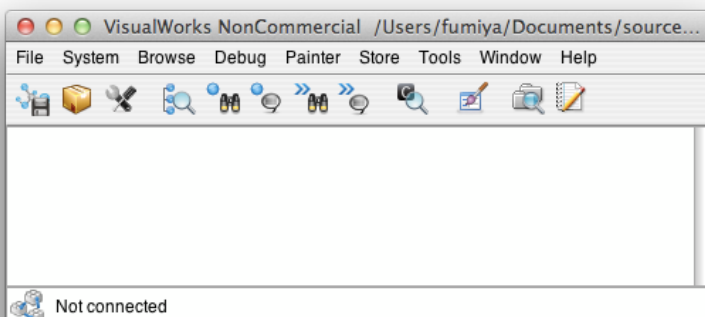


Fix Colored Text Selection を選択

これから、この状態で image を保存するが、Transcript に残骸が残っているのは嫌なので、ちょっと処理をする



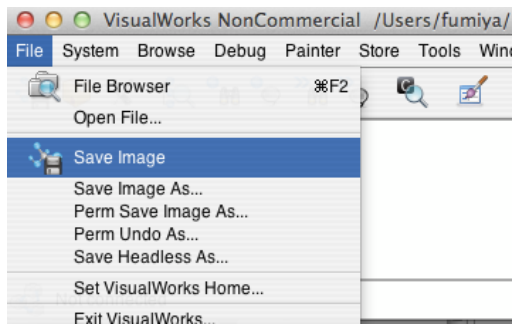
Do it すると



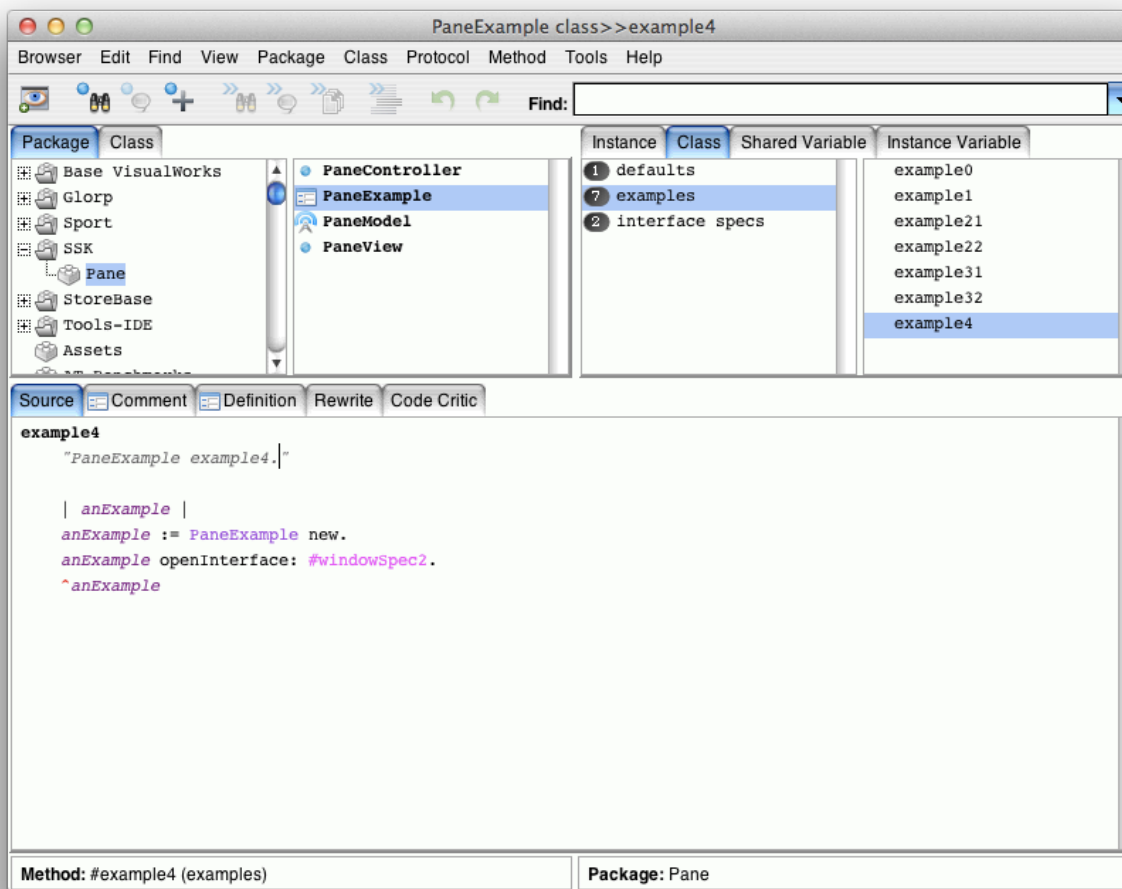
この通りきれいに

この状態で Image を保存することに

これからは、毎回ロードをしなくてもコードがカラーリングされるようになる



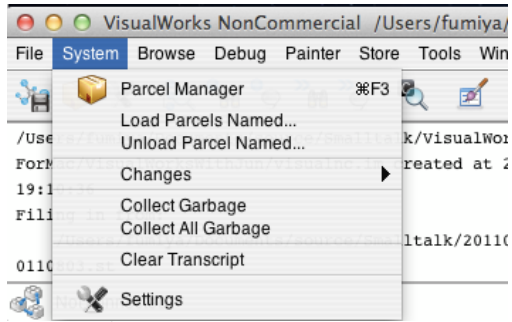
これで開発環境はそろったので
いつも通り File Browser を開いて File in



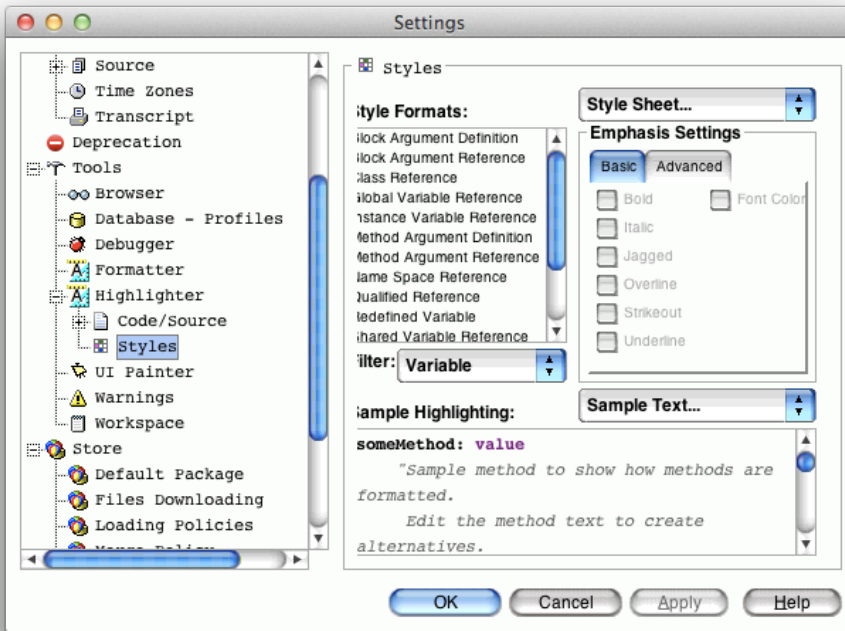
こんな感じでカラーリングされてる！

書いている途中でもカラーリングをきれいに変えてくれるので、確保していない Temporary 変数に関して怒られるので、気になるようならちゃんと自分で書くこと

フォントの関係で Windows だと斜体の文字が読みにくいらしい…
設定を変えよう

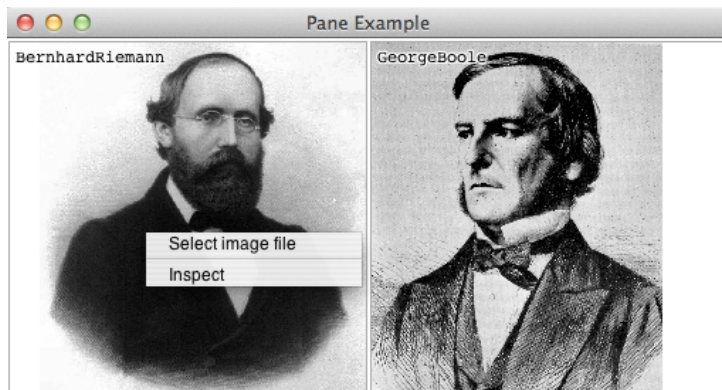


Settings を選択



Style Formats の左端が切れてるけど、バグらしい(頑張れば自分で直せるかな?)
ありとあらゆる Italic をオフにしておけば OK

ちょっと復習

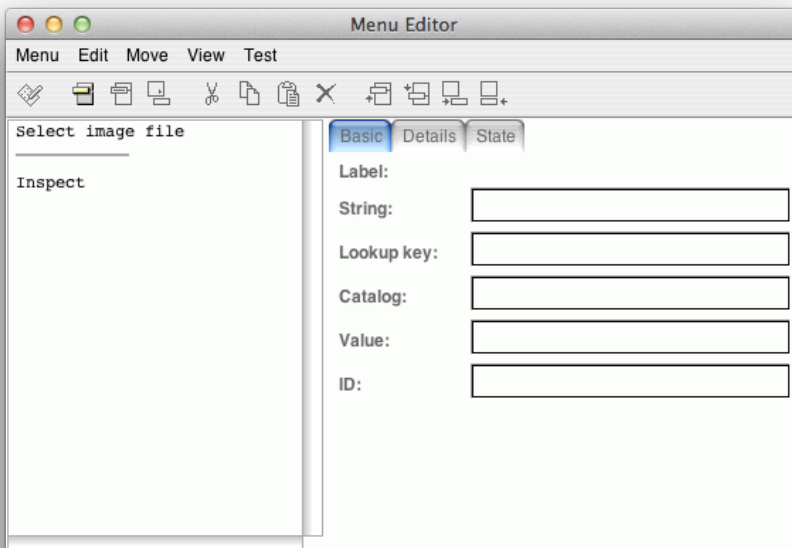
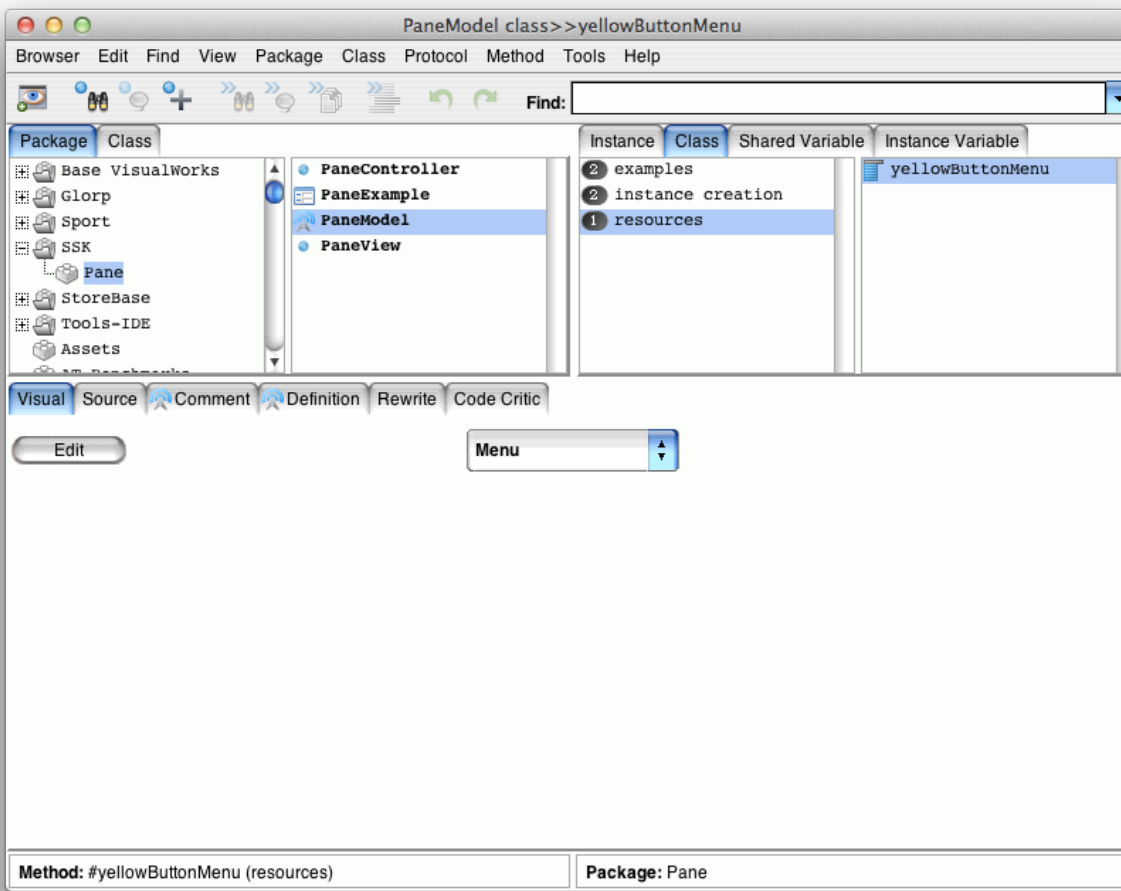


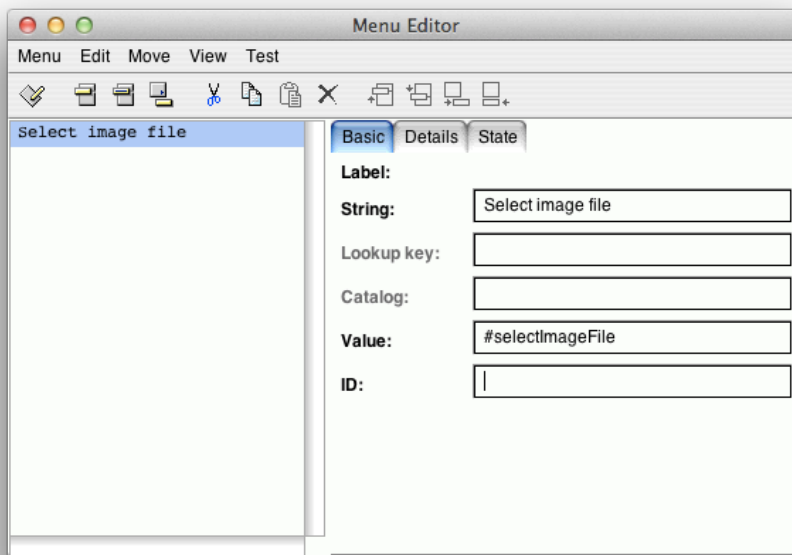
メニューを誰が何を持つかが問題になっていた

PaneModel, Class, resources, yellowButtonMenu
nameKey に持っておけばうまくいきそうという話だった

value にブロックで書けば良いよねという話がメーリングリストに流れていたのをそちらを採用することに

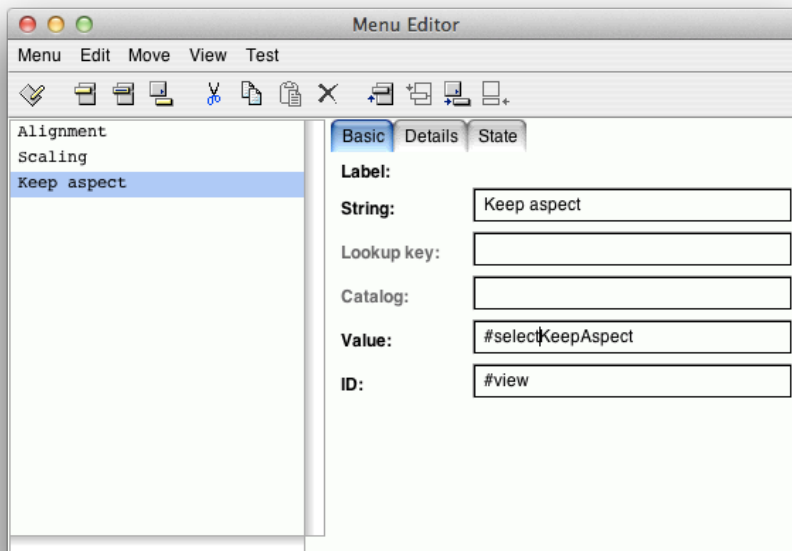
Edit を押して Menu Editor を開く





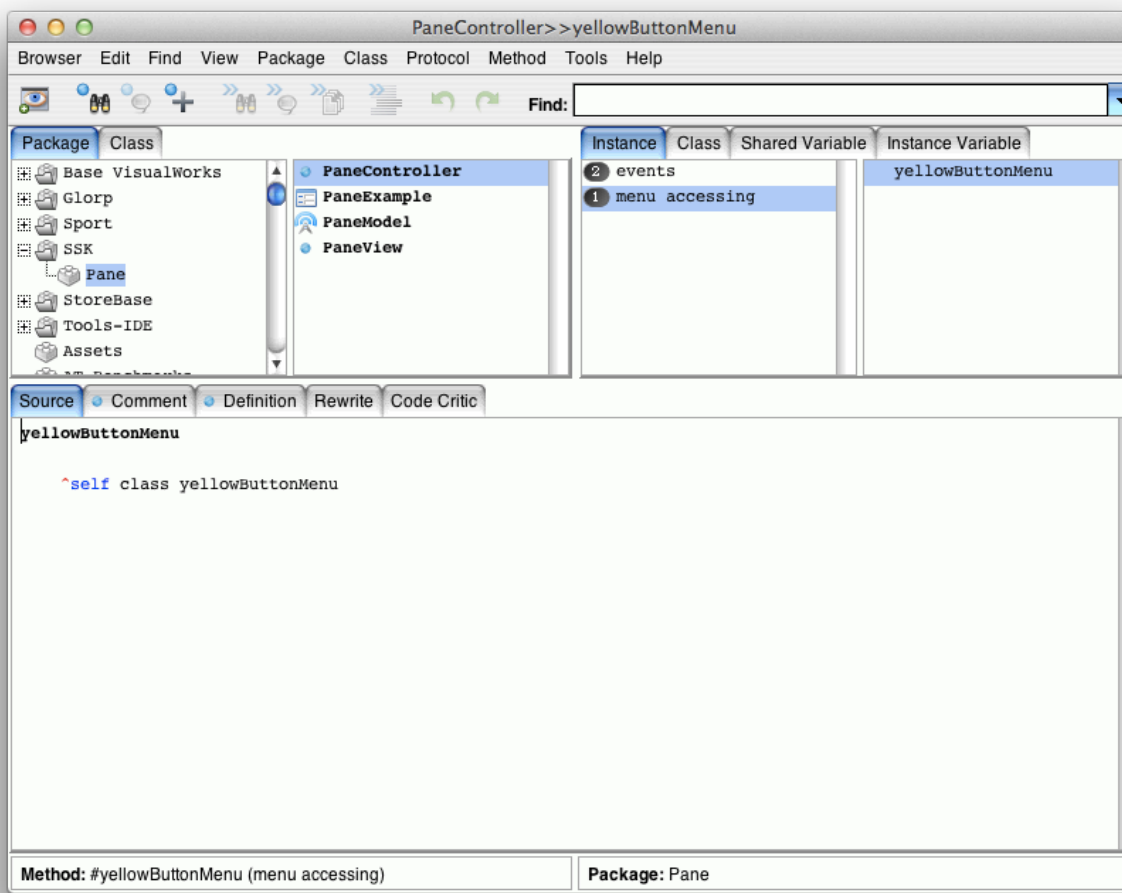
Select image file を残して
ID の部分を削除したのち Menu から Install

Controller も同様に開いて



Keep aspect の Value の select の綴りが間違っていたのを修正
同様に Install

PaneController に新しいカテゴリ menu accessing を追加
PaneModel の menu accessing からまるっとコピー



PaneController, Instance, events, yellowButtonPressedEvent
 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.
aMenu addPart: self yellowButtonMenu.
aSymbol := aMenu startUp.
(aSymbol isKindOfClass: Symbol) ifTrue: [self model perform: aSymbol].
^nil

```



こんな感じでメニューが増えた！

PaneModel, Instance, menu accessing, yellowButtonMenu
 yellowButtonMenu

```

I aMenu I
aMenu := self class yellowButtonMenu.
aMenu menuItems do:
  [:anItem I
  I aSymbol I
  aSymbol := anItem value.
  (aSymbol isKindOfClass: Symbol)
  ifTrue:
    [I aBlock I
    aBlock := [self perform: aSymbol].
    anItem value: aBlock]].
^aMenu

```

Controller 側も同じように編集していく
PaneController, menu accessing, yellowButtonPressedEvent

yellowButtonMenu

```
| aMenu |
aMenu := self class yellowButtonMenu.
aMenu menuItems do:
    [:anItem |
    | aSymbol |
    aSymbol := anItem value.
    (aSymbol isKindOf: Symbol)
    ifTrue:
        [| aMessageSend |
        aMessageSend := MessageSend receiver: self selector: aSymbol.
        anItem value: aMessageSend]].
^aMenu
```

ところで、MessageSend って何だろう？

workspace でお試し

```
[3 perform: #sqrt] value 1.73205
(MessageSend receiver: 3 selector: #sqrt) value 1.73205
例えばこんな感じで使う
```

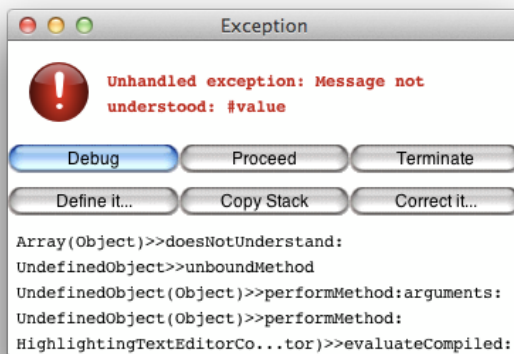
ここで出てきた value メッセージはいろいろ使えるんだけど…

```
#sqrt value #sqrt
```

以下の2種類ではうまく動かない…

```
 #(10 20 30) value
Object new value
```

Exception…



なので、valueをやめて evaluate を使ってみよう

```
[3 perform: #sqrt] evaluate 1.73205
(MessageSend receiver: 3 selector: #sqrt) evaluate 1.73205
#sqrt evaluate #sqrt
```

value ではダメだったものでも…

```
 #(10 20 30) evaluate #(10 20 30)
Object new evaluate an Object
```

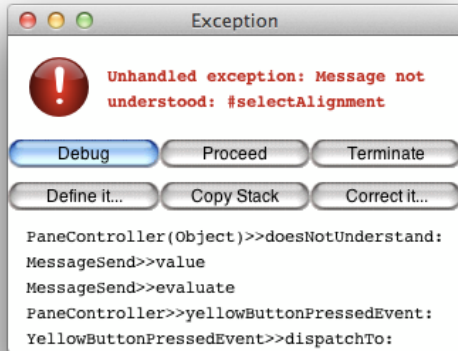
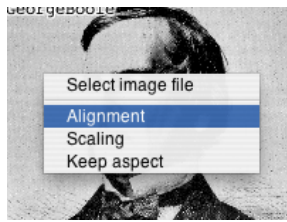
evaluate だと全部ちゃんと返事してくれるので、こちらを使うことにしましょう
ということで yourself と evaluate を覚えておきましょう (自分自身と評価結果を返す)

PaneController, Instance, events, yellowButtonPressedEvent

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

```
| aMenu |
aMenu := self model yellowButtonMenu.
aMenu addPart: self yellowButtonMenu.
aMenu startUp evaluate.
^nil
```

例えば、Alignment を選んでみると



Exception が出るけど、ちゃんと、Controller が受信するようになっている

とりあえず、Transcript でお試し

PaneController, Instance, menu messages, selectAlignment
selectAlignment

```
Transcript
cr;
show: thisContext printString
```

全く同じ内容で、selectKeepAspect, selectScaling を追加

selectScaling と selectKeepAspect はトグルスイッチとして実装することに
PaneController, Instance, menu messages, selectScaling
selectScaling

```
self view isScaling: self view isScaling not.
self model changed
```

PaneController, Instance, menu messages, keepAspect
selectKeepAspect

```
self view keepAspect: self view keepAspect not.
self model changed
```

トグルスイッチでは実装できない Alignment 位置をどう変更していくか？

PaneView, Instance, accessing, alignmentSymbol:
をまるまるコピーしたものを書くとコードクローンが出来てしまう(修正が複数箇所に及ぶ)ので…

PaneView, Instance, accessing alignmentSymbols を追加
alignmentSymbols

```
^#(#topLeft #topCenter #topRight #leftCenter #center #rightCenter #bottomLeft #bottomCenter #bottomRight #origin #corner)
```

定数(リテラル)を返すようにして、定数を外部に置くことにする
プログラマ屋さんの書き方

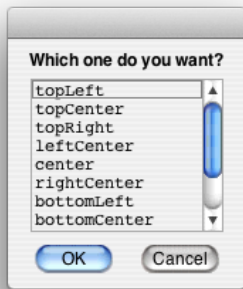
PaneView, Instance, accessing alignmentSymbol:
alignmentSymbol: aSymbol

```
(self alignmentSymbols includes: aSymbol) ifFalse: [^nil].
alignmentSymbol := aSymbol
```

PaneController, Instance, menu messages, selectAlignment
selectAlignment

```
| aSymbol |
aSymbol := Dialog
    choose: 'Which one do you want?'
    fromList: self view alignmentSymbols
    values: self view alignmentSymbols
    lines: 8
    cancel: [^nil].
```

```
self view alignmentSymbol: aSymbol.  
self model changed
```



こんなものが出るようになる

PaneModel, Instance, menu messages selectImageFile
selectImageFile

```
! aFilename !  
aFilename := Dialog requestFileName: 'Which file do you want?'.  
aFilename ifNil: [^nil].  
^aFilename "self でも動く"
```

PaneModel の Instance に private カテゴリを追加
filename: aFilename

```
! anImage !  
anImage := JunImageUtility fromFile: aFilename.  
self picture: anImage.  
self label: (Filename splitExtension: aFilename tail) first
```

PaneModel, Clas, instance creation, filename:
filename: aFilename

```
! aModel !  
aModel := self new.  
aModel filename: aFilename.  
^aModel
```

PaneModel, Instance, menu messages selectImageFile
selectImageFile

```
! aFilename !  
aFilename := Dialog requestFileName: 'Which file do you want?'.  
aFilename ifNil: [^nil].  
self filename: aFilename.  
self changed.  
^aFilename
```

と編集したが、ファイルではなくファイル名が返ってきたので

selectImageFile

```
! aString aFilename !  
aString := Dialog requestFileName: 'Which file do you want?'.  
aString ifNil: [^nil].  
aFilename := aString asFilename.  
aFilename ifNil: [^nil].  
self filename: aFilename.  
self changed.  
^aFilename
```

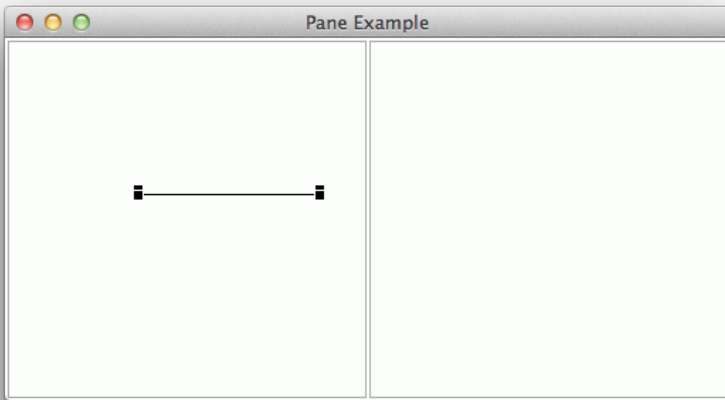
ここで最低限のものは完成

これからは機能拡張

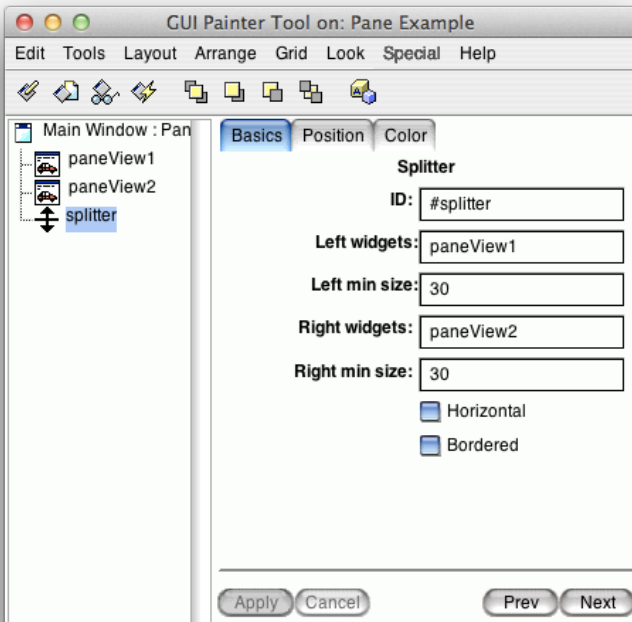
画像間のセパレータを動かせるようにする

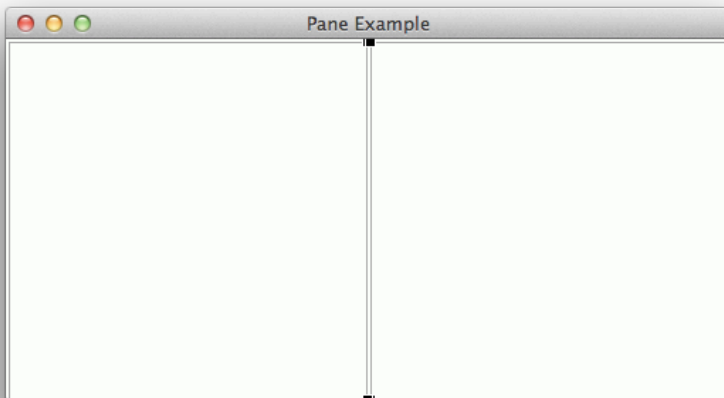
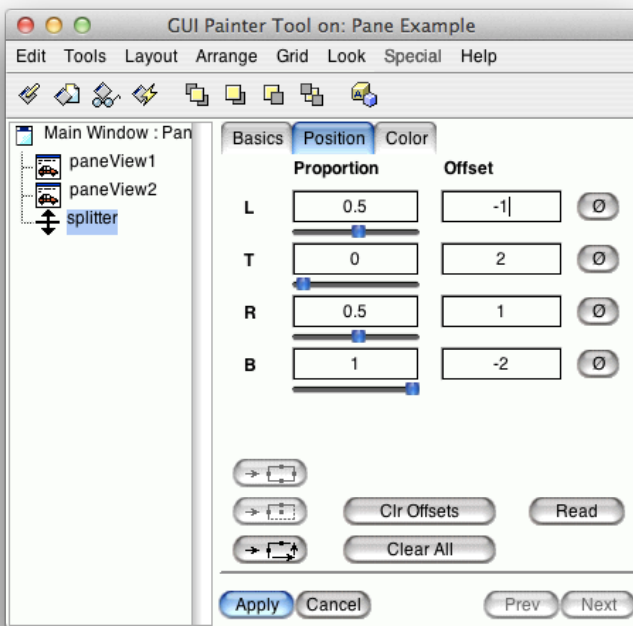
PaneExample, Class, interface specs, windowSpec2
Edit をクリックして



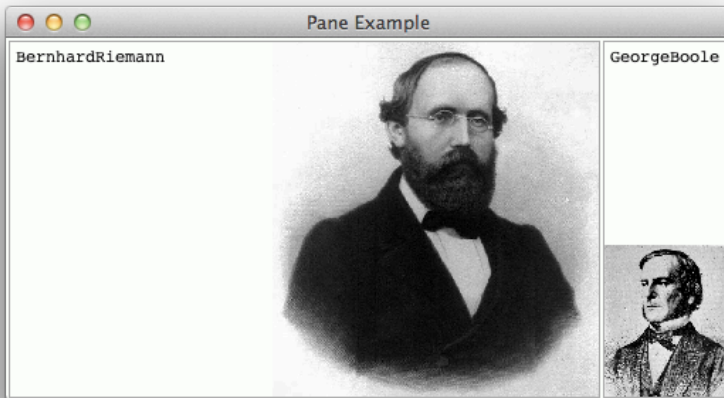


適当に置いて



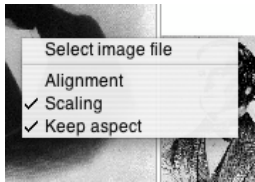


このようになる
Install して、example4 から再起動(さすがに再起動が必要らしい...)



こんなことが出来るように

こんな感じで



✓が出るようにしたい

PaneController, Instance, menu accessing, yellowButtonMenu
yellowButtonMenu

```

I aMenu I
aMenu := self class yellowButtonMenu.
aMenu menuItems do:
    [:anItem I
    | aSymbol I
    aSymbol := anItem value.
    (aSymbol isKindOf: Symbol)
    ifTrue:
        [I aMessageSend I
        aMessageSend := MessageSend receiver: self selector: aSymbol.
        anItem value: aMessageSend].
    anItem label = 'Scaling'
    ifTrue: [self view isScaling ifTrue: [anItem beOn] ifFalse: [anItem beOff]].
    anItem label = 'Keep aspect'
    ifTrue: [self view keepAspect ifTrue: [anItem beOn] ifFalse: [anItem beOff]].
^aMenu

```

このままだとメニューの名前が変わる(日本語にしたときとか)と動かないので、シンボルで動く様に

yellowButtonMenu

```

I aMenu I
aMenu := self class yellowButtonMenu.
aMenu menuItems do:
    [:anItem I
    | aSymbol I
    aSymbol := anItem value.
    (aSymbol isKindOf: Symbol)
    ifTrue:
        [I aMessageSend I
        aSymbol = #selectSaling
        ifTrue: [self view isScaling ifTrue: [anItem beOn] ifFalse: [anItem beOff]].
        aSymbol = #selectKeepAspect
        ifTrue: [self view keepAspect ifTrue: [anItem beOn] ifFalse: [anItem beOff]].
        aMessageSend := MessageSend receiver: self selector: aSymbol.
        anItem value: aMessageSend]].
^aMenu

```

PaneController, Instance, menu messages, selectAlignment
を頭に "_" をつけてバックアップ

_selectAlignment

```

I aSymbol I
aSymbol := Dialog
    choose: 'Which one do you want?'
    fromList: self view alignmentSymbols

```

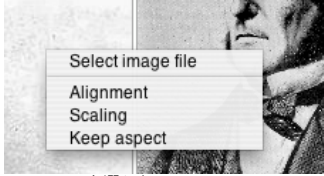
```

values: self view alignmentSymbols
lines: 8
cancel: [^nil].
self view alignmentSymbol: aSymbol.
self model changed

```

PaneController, Instance, menu messages, selectAlignment
selectAlignment

(Menu labelArray: self view alignmentSymbols) startUp
こんな感じに変更



Alignment を選ぶと



こんなものが出るように変更できた

こちらも✓が入るように変更

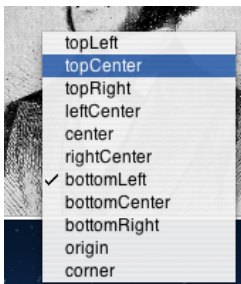
selectAlignment

```

I anArray aSymbol aMenu I
anArray := self view alignmentSymbols.
aMenu := Menu labelArray: anArray values: anArray.
aMenu menuItems do:
    [:anItem I
        anItem value = self view alignmentSymbol
            ifTrue: [anItem beOn]
            ifFalse: [anItem beOff]].
aSymbol := aMenu startUp.
(aSymbol isKindOfClass: Symbol)
    ifTrue:
        [self view alignmentSymbol: aSymbol.
         self model changed]

```

✓が入るようになった



実はちょっとバグがあったので修正

PaneModel, Instance, menu messages, selectImageFile
selectImageFile

```

I aString aFilename I
aString := Dialog requestFileName: 'Which file do you want?'.
aString ifNil: [^nil].
aString isEmpty ifTrue: [^nil].
aFilename := aString asFilename.
aFilename ifNil: [^nil].
self filename: aFilename.
self changed.
^aFilename

```