CS 683 Emerging Technologies: Embracing Change
Spring Semester, 2001
Doc 19 Perform & Menu
Contents

References..... 1
Symbols 2
Simple Dialog..... 4
Popup Menus..... 5

References

Squeak: Object-Oriented Design with Multimedia Applications, Guzdial, 2001, Chapter 3, Chapter 5

Squeak Source Code

Copyright©, All rights reserved. 2001 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (http://www.opencontent.org/opl.shtml) license defines the copyright on this document.

Symbols

A symbol is formed by:

- starting a character sequence with #

  #aSymbol
  #ANOTHERSymbol

- Sending asSymbol to a string

  'cat' asSymbol

Symbols are created uniquely

Only one instance of a symbol with a given character sequence will exist in the image

The following pointer compare is false

```
'cat' == 'tac' reversed
```

The following pointer compare is true

```
'cat' asSymbol == 'tac' reversed asSymbol
```

perform: aSymbol

Object implements perform: aSymbol

perform: aSymbol
aSymbol must be a unary method name in the receiver
Execute the unary method in the receiver

perform: aSymbol with: anObject
aSymbol must be an one argument method in the receiver
Execute the method with anObject as the argument

perform: aSymbol with: arg1 with: arg2
perform: aSymbol with: arg1 with: arg2 with: arg3
perform: aSymbol withArguments: anArray

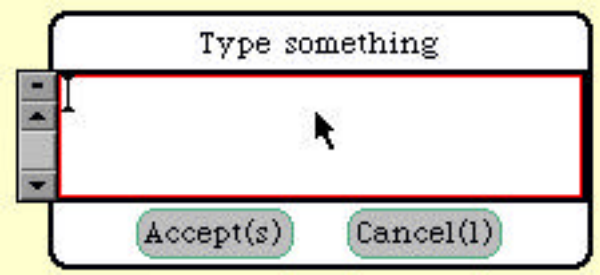
Examples

Action	Result
'cat' perform: #reversed	tac
'cat' perform: #at: with: 1	$c
'cat' perform: 'reversed' asSymbol	'tac'
3 perform: #+ with: 2	5
'smalltalk' perform: #copyFrom:to: with: 6 with: 9	'talk'
'smalltalk' perform: #copyFrom:to: withArguments: #(6 9)	'talk'

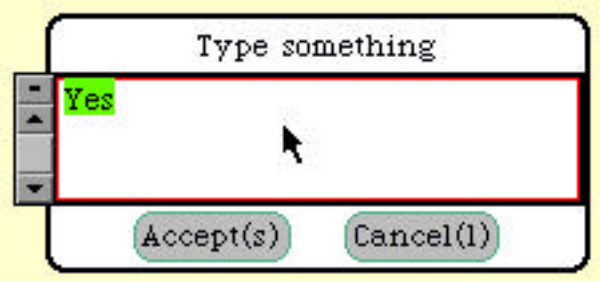
Simple Dialog

FillInTheBlankMorph returns
- The string the user entered into the text field
- Empty string returned when user cancels

FillInTheBlankMorph request: 'Type something'



FillInTheBlankMorph
request: 'Type something'
initialAnswer: 'Yes'



FillInTheBlankMorph requestPassword: 'Password?'

Popup Menus

Menus have

- Optional Title
- Menu items
- Submenus
- Lines separating items
- Stay up item
- Optional Default target for menu items

Menu Items have

- Display text
- Target

  Object to be notified when menu item is selected

- Selector (action)

  Methods sent to target when menu item is selected

- Argument(s) to selector

  If selector has one or more arguments,
  Then must store them with the menu item

Some Menu methods
Instance Creation

MenuMorph new
MenuMorph entitled: 'Popup'

Instance method

add: aDisplayString target: anObject selector: aSymbol
add: aString target: anObject selector: aSymbol argument: anObject
add: aString target: target selector: aSymbol argumentList: array

add: aString subMenu: aMenu.

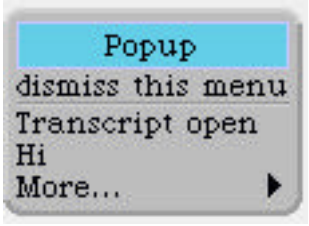
add: aString action: aSymbol
add: aString selector: aSymbol
add: aString selector: aSymbol argument: anObject
    aSymbol is sent to the default target

defaultTarget: anObject

position: aPoint
    aPoint becomes the upper left corner of the menu

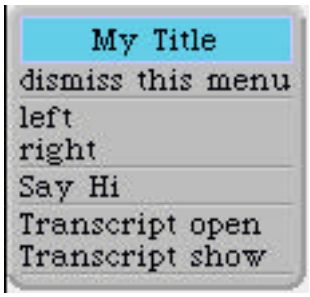
addStayUpItem

Simple Example




```
| menu nestedMenu |
nestedMenu := MenuMorph new.
nestedMenu
   add: 'Bye'
   target: Transcript
   selector: #show:
   argument: 'bye'.
menu := MenuMorph entitled: 'Popup'.
menu addStayUpItem.
menu
   add: 'Transcript open'
   target: Transcript
   selector: #open.
menu
   add: 'Hi'
   target: Transcript
   selector: #show:
   argument: 'hi'.
menu
   add: 'More...'
   subMenu: nestedMenu.
menu openInWorld
```

Example with Real Target




```
| menu |
menu := MenuMorph entitled: 'My Title'.
menu
   addStayUpItem;
   defaultTarget: MenuModelExample new.
menu
   add: 'left'
   action: #left.
menu
   add: 'right'
   action: #right.
menu addLine.
menu addLine.  "extra lines ignored"
menu
   add: 'Say Hi'
   selector: #displayMessage:
   argument: 'hi'.
menu addLine.
menu
   add: 'Transcript open'
   target: Transcript
   action: #open.
menu
   add: 'Transcript show'
   target: Transcript
   selector: #show:
   argument: 'hi'.
menu addLine.
menu openInWorld
```

MenuModelExample

```
Object subclass: #MenuModelExample
   instanceVariableNames: ''
   classVariableNames: ''
   poolDictionaries: ''
   category: 'Whitney-Examples'

displayMessage: aString
   Transcript
      show: aString;
      cr.! !

left
   self displayMessage: 'left called'

right
   self displayMessage: 'right called'
```