## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Morphic</td>
<td>2</td>
</tr>
<tr>
<td>Morphs</td>
<td>3</td>
</tr>
<tr>
<td>Keyboard Events</td>
<td>5</td>
</tr>
<tr>
<td>Handling Mouse Events</td>
<td>7</td>
</tr>
<tr>
<td>Drawing</td>
<td>9</td>
</tr>
<tr>
<td>Stepping</td>
<td>10</td>
</tr>
</tbody>
</table>

## Reference

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

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

New interface experiment from Self

Goals were:

• Concreteness
  Language supports
  Direct, copy & modify style of programming
  Interface supports
  Immediate & direct access to all parts of application even when running

• Uniformity
  Language
  Merges state and behavior
  Uses objects & messages for everything
  Interface
  Uses graphical objects down to the lowest level
  Removes the distinction between run and edit

• Flexibility
Morphs

Live an a world

Can interact with
  • Its world
  • Other morphs in the world

Can be programmed with:
  • Smalltalk code
  • Tiles and scripts
  • Graphical wired together (Fabrik)
Basic Interaction

Simple Events

Keyboard
Mouse
Drag & Drop
Keyboard Events

Three types in order sent

- Key down
- Key stroke
  Contains key pressed
- Key up

To handle keyboard events Morph subclass must implement:

```plaintext
dehandlesKeyboard: anKeyboardEvent
```

The method must return true for the types of events of interest

**Important KeyboardEvent Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>isKeyDown</code></td>
<td>anyModifierKeyPressed</td>
</tr>
<tr>
<td><code>isKeyUp</code></td>
<td>commandKeyPressed</td>
</tr>
<tr>
<td><code>isKeyboard</code></td>
<td>controlKeyPressed</td>
</tr>
<tr>
<td><code>isKeystroke</code></td>
<td>macOptionKeyPressed</td>
</tr>
<tr>
<td><code>keyCharacter</code></td>
<td>shiftedPressed</td>
</tr>
<tr>
<td><code>keyValue</code></td>
<td></td>
</tr>
</tbody>
</table>
To Actually handle Keyboard Events

Morph subclass must implement

keyDown: anEvent
keyStroke: anEvent
keyUp: anEvent
Handling Mouse Events

A Morph subclass must override one or more of the following methods to be notified of the event:

Return true to handle the event:

- `handlesKeyboard: aMouseButtonEvent`
- `handlesMouseDown: aMouseButtonEvent`
- `handlesMouseOver: aMouseButtonEvent`
- `handlesMouseOverDragging: aMouseButtonEvent`
- `handlesMouseStillDown: aMouseButtonEvent`

To handle the event implement one of:

- `mouseDown: aMouseButtonEvent`
- `mouseEnter: aMouseButtonEvent`
- `mouseEnterDragging: aMouseButtonEvent`
- `mouseLeave: aMouseButtonEvent`
- `mouseLeaveDragging: aMouseButtonEvent`
- `mouseMove: aMouseButtonEvent`
- `mouseStillDown: aMouseButtonEvent`
- `mouseStillDownThreshold aMouseButtonEvent`
- `mouseUp: aMouseButtonEvent`
## Important MouseButtonEvent Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>isDraggingEvent</td>
<td>anyButtonPressed</td>
</tr>
<tr>
<td>isMouse</td>
<td>blueButtonPressed</td>
</tr>
<tr>
<td>isMouseDown</td>
<td>cursorPoint</td>
</tr>
<tr>
<td>isMouseEnter</td>
<td>redButtonPressed</td>
</tr>
<tr>
<td>isMouseLeave</td>
<td>targetPoint</td>
</tr>
<tr>
<td>isMouseMove</td>
<td>yellowButtonPressed</td>
</tr>
<tr>
<td>isMouseUp</td>
<td></td>
</tr>
<tr>
<td>isMove</td>
<td></td>
</tr>
</tbody>
</table>
Drawing

To change the image of the Morph subClass

Override the method

drawOn: aCanvas
Stepping

Morphs are active

Periodically they are sent the message step

This method can perform animation etc.

The time between sending step is determined by the method

    stepTime

This method returns time in milliseconds

To start stepping send:

    startStepping

To send stepping send:

    endStepping
Example

Morph subclass: #TestMorphic
  instanceVariableNames: ""
  classVariableNames: ""
  poolDictionaries: ""
  category: 'Whitney-Morphic'!

step
  self world bounds bottom > self bottom
    ifTrue:[self bottom: (self bottom + 3)]

stepTime
  ^500

handlesKeyboard: aKeyBoardEvent
  ^aKeyBoardEvent isKeystroke

keyStroke: anEvent
  Transcript
    show: 'You pressed key ';
    show: anEvent keyCharacter;
    cr.

drawOn: aCanvas
  aCanvas
    fillOval: self bounds
    color: Color red