Joe the Box

• Joe the Box is a little bit of history
  — first invented by Adele Goldburg for Smalltalk 72
  — Alan Kay used it in a 1977 *Scientific American* article

• Two roles:
  — a microworld for exploring object interaction
  — an interesting programming experience

Creating a box

joe := Box new.

1. sends the message new to the class Box, which answers a new Box object.
2. names this new object “joe”

Talking to joe

• Joe understands various messages:
  joe class.
    › answers the kind of object that joe is
  joe show.
    › makes joe visible on the display
  joe turn: 30.
    › joe turns 30 degrees clockwise
Joe can move to a given Point
joe moveTo: 20@30.

So, joe can follow the mouse
[Sensor anyButtonPressed] whileFalse:
[joe moveTo: Sensor peekMousePt].

“Sensor” is the mouse or pointer
[ ] takes code and turns it into an object
whileFalse does what it claims!

You can make many boxes:
jill := Box new

Joe and jill are independent
jill grow: 20.
joe turn: 50.

The Box protocol

Boxes understand the following messages:

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show</td>
<td>draws the box on the display</td>
</tr>
<tr>
<td>hide</td>
<td>erases the box from the display (but it still exists)</td>
</tr>
<tr>
<td>move: aPoint</td>
<td>moves by the increment expressed by aPoint</td>
</tr>
<tr>
<td>moveTo: aPoint</td>
<td>moves the box to aPoint</td>
</tr>
<tr>
<td>grow: n</td>
<td>expands the box by n pixels; negative n shrinks</td>
</tr>
<tr>
<td>turn: degrees</td>
<td>rotates the box by degrees</td>
</tr>
</tbody>
</table>

The Box class protocol

The Box class understands the following messages:

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td>creates a new Box and answers it</td>
</tr>
<tr>
<td>allInstances</td>
<td>answers a collection of all of the Boxes that currently exist</td>
</tr>
</tbody>
</table>
Getting the Box code

- As a changeset from the class website
  - drop the Boxes.cs file onto your running Pharo image, and select “install into new changeset”
- As a package from SqueakSource
  1. add this repository in Monticello:
     ```plaintext
     MCHttpRepository
     location: 'http://www.squeaksource.com/PSUCS520'
     user: '{your Squeaksource user name}'
     password: '{your password}'
     
     2. Load the newest version of CS520-Boxes
     ```

Example

- Create a new class `DigitalClock`:
  ```plaintext
  TextMorph subclass: #DigitalClock
  instanceVariableNames: ''
  classVariableNames: ''
  poolDictionaries: ''
  category: 'CS520'
  
  give it a step method:
  ```
  ```plaintext
  step
  self newContents: Time now asString
  
  create an object and display it
  d := DigitalClock new openInWorld
  
  Try the effect of `d startStepping` and `d stopStepping`
  ```

Random numbers

- anInterval atRandom
  - answers a number chosen at pseudo-random from anInterval
  - e.g., `(1 to: 5) atRandom` answers 1, 2, 3, 4, or 5
- more sophisticated pseudo-random numbers can be obtained using the class `Random`
  - read the class comment for `Random`

Stepping

- To help make animations, Pharo provides the following protocol for Morphs (i.e., displayable objects)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>step</td>
<td>the step message is sent to an object periodically by the display. So, any code that you write there will be executed.</td>
</tr>
<tr>
<td>startStepping</td>
<td>turn on the periodic step messages</td>
</tr>
<tr>
<td>stopStepping</td>
<td>turn off the periodic step messages</td>
</tr>
<tr>
<td>stepTime</td>
<td>the interval between step messages; the default is 1000 (milliseconds). Override this method to change the step interval. (Look at Morph ›› step)</td>
</tr>
</tbody>
</table>

Random numbers

- anInterval atRandom
  - answers a number chosen at pseudo-random from anInterval
  - e.g., `(1 to: 5) atRandom` answers 1, 2, 3, 4, or 5
- more sophisticated pseudo-random numbers can be obtained using the class `Random`
  - read the class comment for `Random`