Object-Oriented GUIs

Andrew P. Black

Portland State UNIVERSITY

Object-Oriented GUIs

- In the beginning, there was Chaos
- And then, there was MVC
  - Model
  - View
  - Controller

Model–View–Controller

- The key idea is to separate the application logic — the Model — from
- The View — one or more visualizations of the model on the display, and
- The Controller — which handles user input on the view, and causes the model to change in response.

Why MVC?

- Manage complexity
- Re-use models with different views
- Re-use views with different models

Warning: MVC ≠ MVC

- There used to be a package, in Squeak, called MVC.
  - It implemented MVC
  - It has been removed from Pharo
- Morphic is the only User-interface framework in Pharo

Morphic

- Morphic is the name of Pharo's UI framework
  - Morph is also the name of the base (abstract) class that implements *Morphs*
  - Morphs combine View and Control
  - The Model can be a separate collection of objects, or the Morphs can be their own model.

Model-View separation

When should you separate Model and View?

Why MVC?

- Manage complexity
- Re-use models with different views
- Re-use views with different models

Model-View separation

When should you separate Model and View?

- if the complexity is high
- if there is a chance for re-use

Dancing Boxes

- joeTheBox is an example of a Morph with its own behavior
- behavior is very simple — no need to separate the graphical part from the behavioral part
- Hence, we gave Morphs application-specific behavior, such as danceWith: