Visual Programming—Common Strategies

- Concreteness - Each visual piece represents a concrete value
- Directness - Manipulate objects or values directly
- Explicitness - Explicitly depict dataflow relationships with edges between objects
- Immediate Visual Feedback - Automatic recalculation of values, etc
Examples

- Imperative Visual Programming by Demonstration
  - End user demonstrates actions
  - Used in graphical editing
  - Creating graphical macros

- Form/Spreadsheet Based Visual Programming
  - Create a form and specify contents
  - Used in spreadsheets

- Dataflow Visual Programming
  - Pass data through various filters and manipulators
  - Like MIT’s Scratch

- Rule-Based Visual Programming
  - User specifies a postcondition on a precondition
  - Specify set of rules in a simulation
Issues

- Abstraction - scaling up support for larger programs
- Language Specification - Grammars for visual languages
- Cognitive Theory - Which VPLs improve humans’ ability to program?
Other topics

“Liveness” - immediacy of feedback during the process of editing a program

- Level 1 - No feedback about a program is provided to the programmer
- Level 2 - Can obtain feedback, but not automatically
- Level 3 - Onscreen values are automatically updated on program edits
- Level 4 - Same as 3, and responds to other events such as clock ticks