

JComboBox

- Drop-down menu
- Formed by an array of objects
- toString
- ActionListener
- `getSelectedItem` - Object

JComboBox code

- `String[] choices = {"Select gender", "Female", "Male"};`
- `JComboBox gender=new JComboBox(choices);`
- `gender.addActionListener`
- `gender.getSelectedItem/Objects/Index`

JRadioButton and ButtonGroups

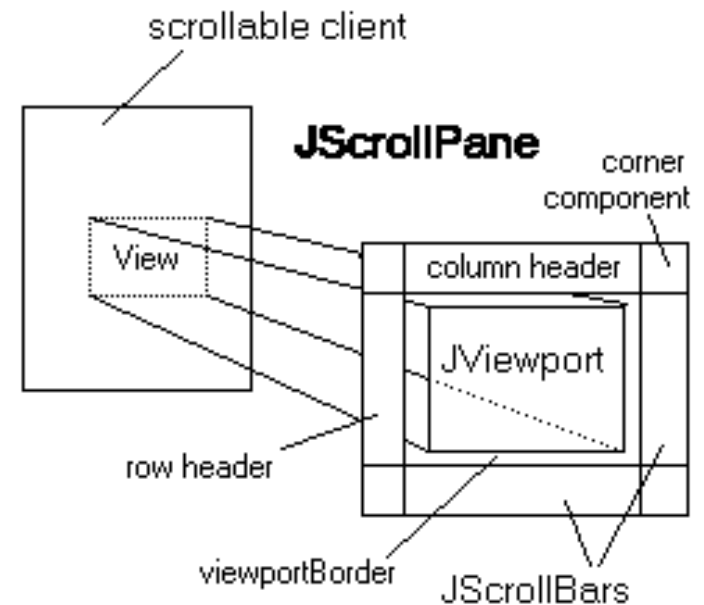
- RadioButtons – each can respond when their dot is pressed
- ButtonGroup
 - Allows for easy organization of items with a “selected” state
 - Turning on one item turns off all others

JRadioButton/ButtonGroup

- `JRadioButton b1 = new JRadioButton("Text");`
- `b1.addActionListener(...`
- `ButtonGroup group = new ButtonGroup();`
- `Group.add(b1);`

JScrollPane

- JScrollPane(Component, vsbPolicy, hsbPolicy)
- Components: JTextArea, ScrollablePicture, JList
- Policies formed in three stages:
 - ScrollPaneConstants.
 - HORIZONTAL_/VERTICAL_
 - AS_NEEDED/NEVER/ALWAYS
- .get/.set for each individual container of pane



JFileChooser

- `.showOpen/SaveDialog(Parent Component)`
- returns `CANCEL_/APPROVE_/ERROR_OPTION` based on which button was selected
- Appears and disappears with parent
- `chooser.getSelectedFile.getName()` – returns full file name – then can use `BufferedReader` (for text files) and `ObjectInput/OutputStream`
- Can filter file types – create a class that extends the abstract class `FileFilter`

Abstract class FileFilter

- Subclass has to overwrite two methods:
- `boolean accept(File f)`: access the name of the file and check what its file type is: `.txt`, `.dat`, etc..., then return a `true/false` based on file type
- `String getDescription()` – return a `String` that lists the acceptable data types

Project 4: Graphical User Interfaces

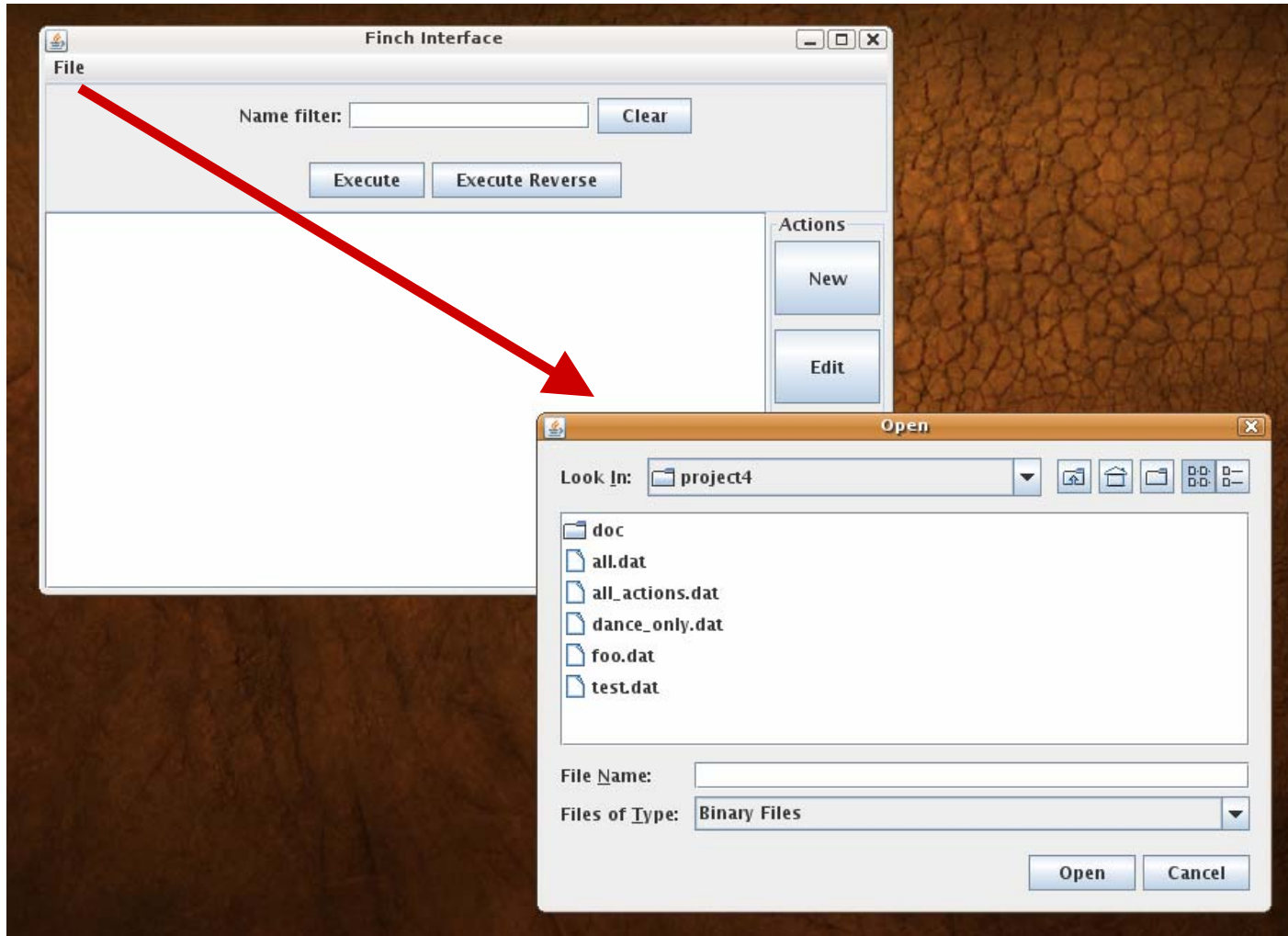
Objective: design and implement a user interface that:

- Involves the interaction of a non-trivial set of graphical components (a Viewer/Controller) with a data store (a Model)
- Employs **Events** and **Listeners** to respond to user behavior
- Opens multiple dialog boxes for short-term user interaction
- Manipulates text and binary files as specified by the user

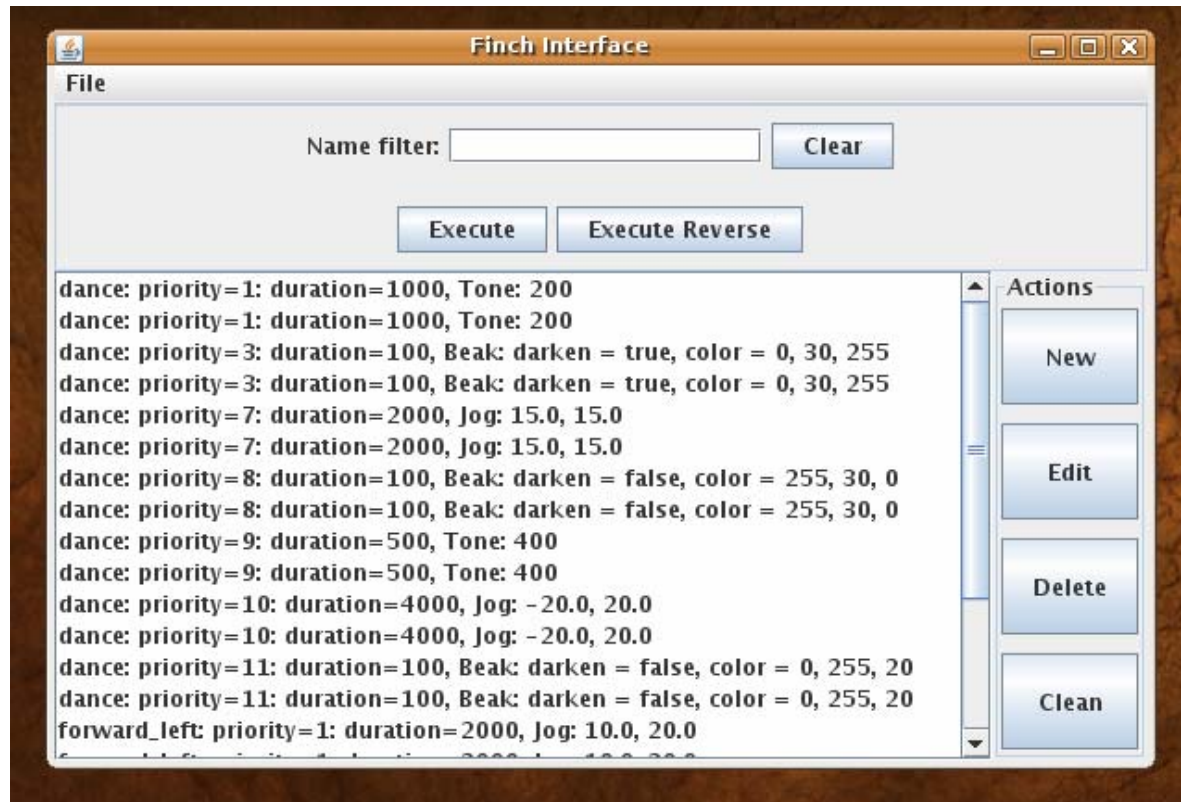
Major Components

- Menu for file loading/merging/saving
- Display and manipulation of an action list: deleting, cleaning, adding and editing
- Dialog box for creating new and editing existing actions
- Dialog box for selecting the nose color
- Execution of action lists

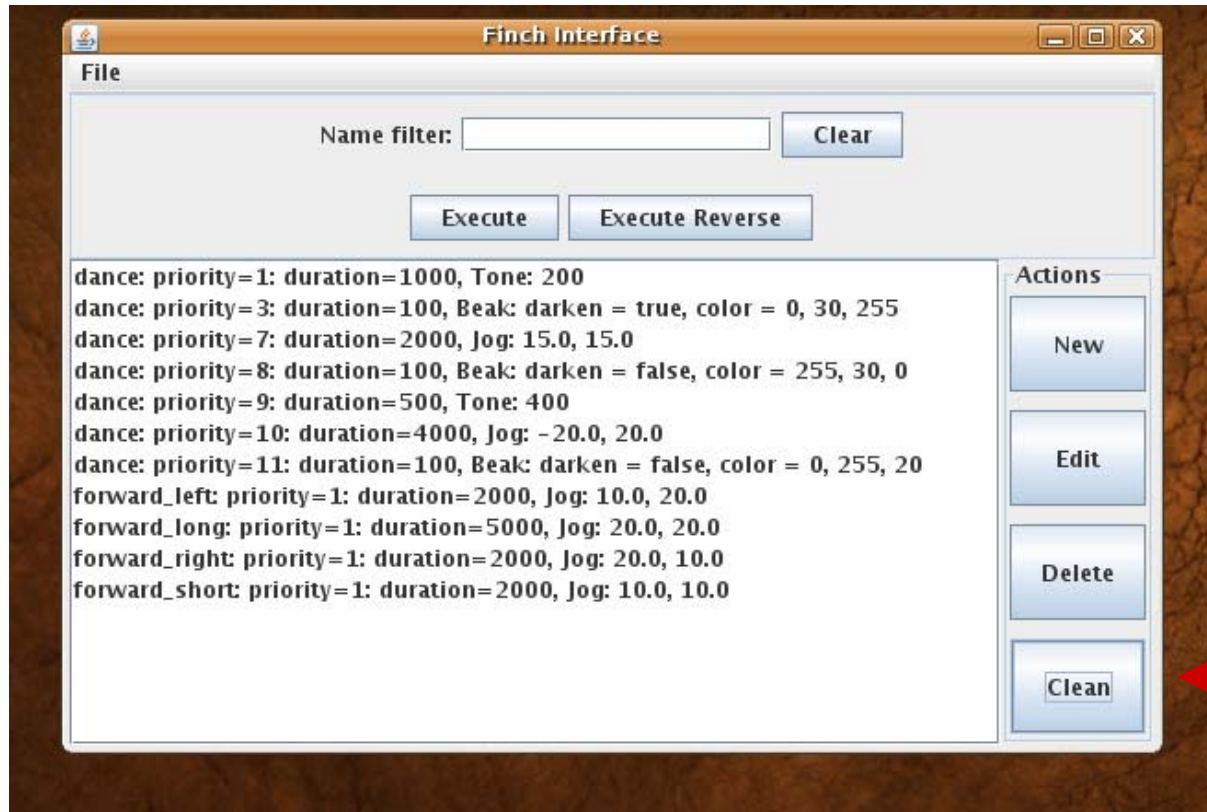
Load Binary File: Opens File Dialog Box



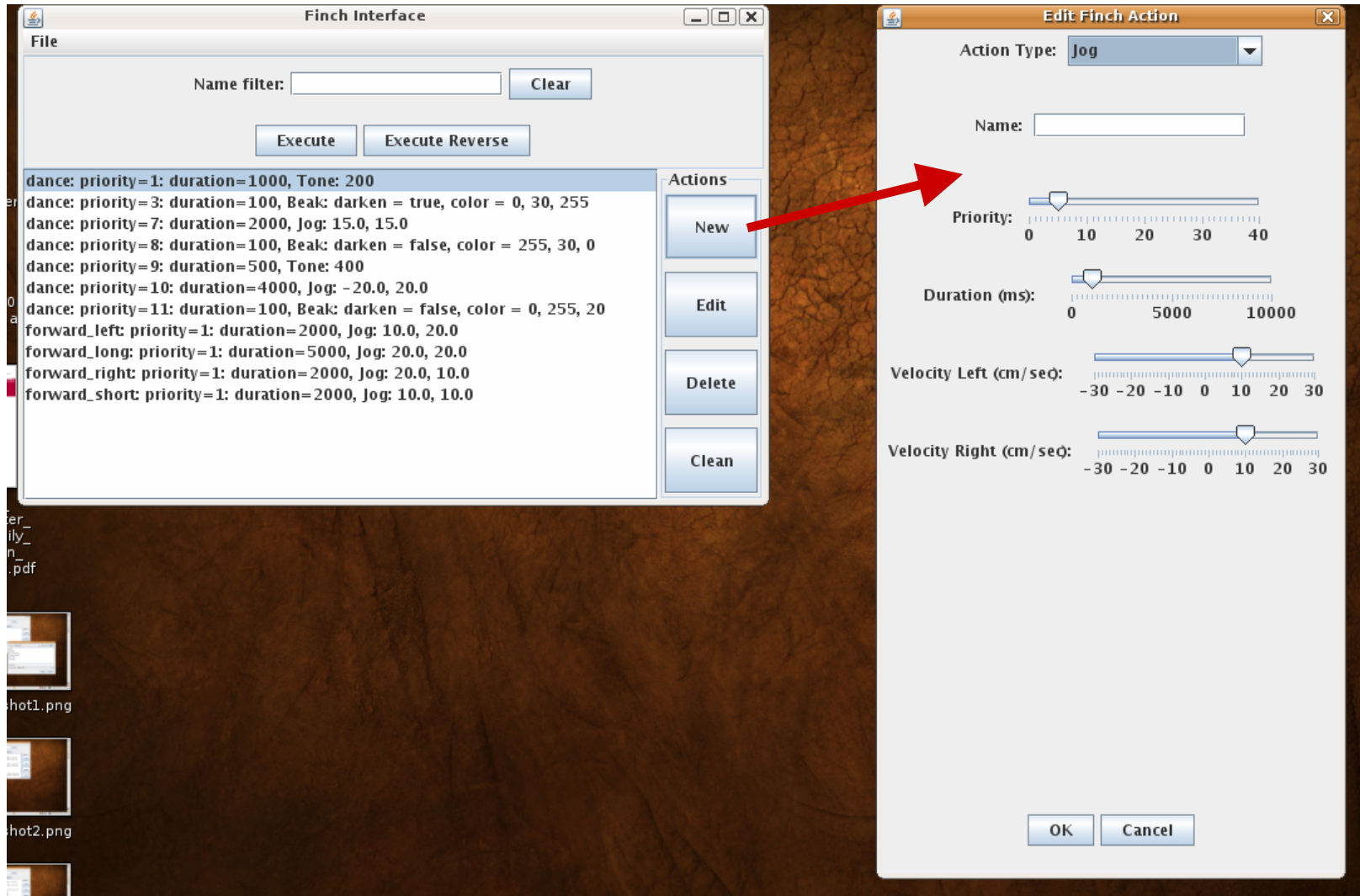
File Loaded



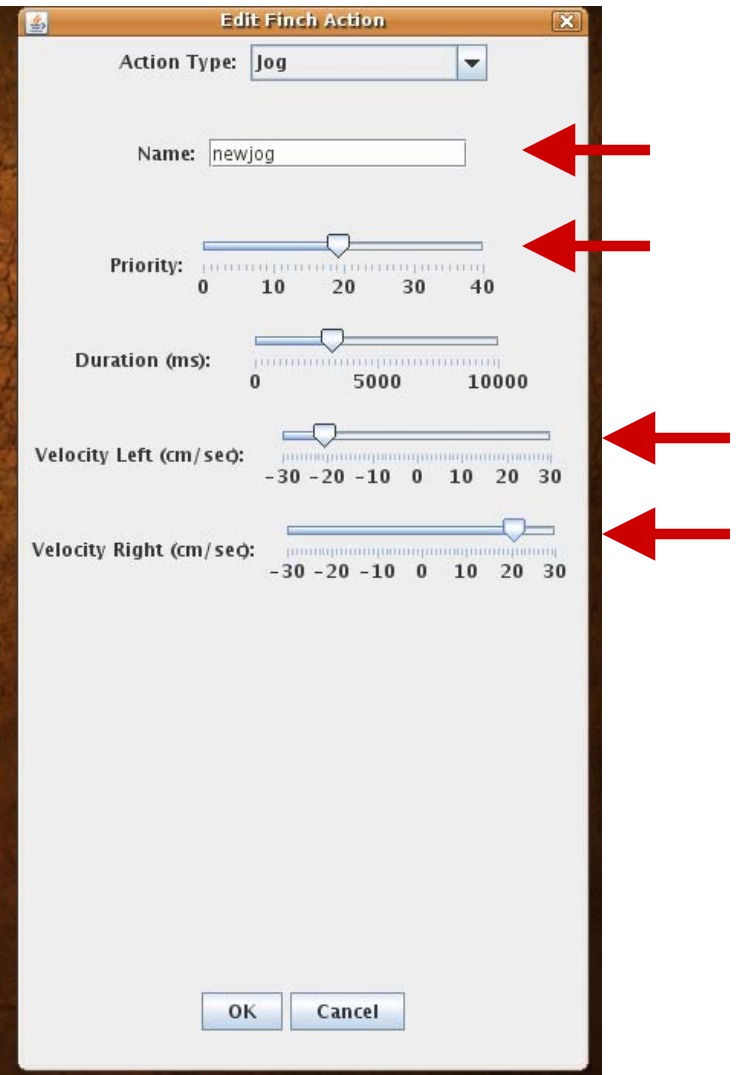
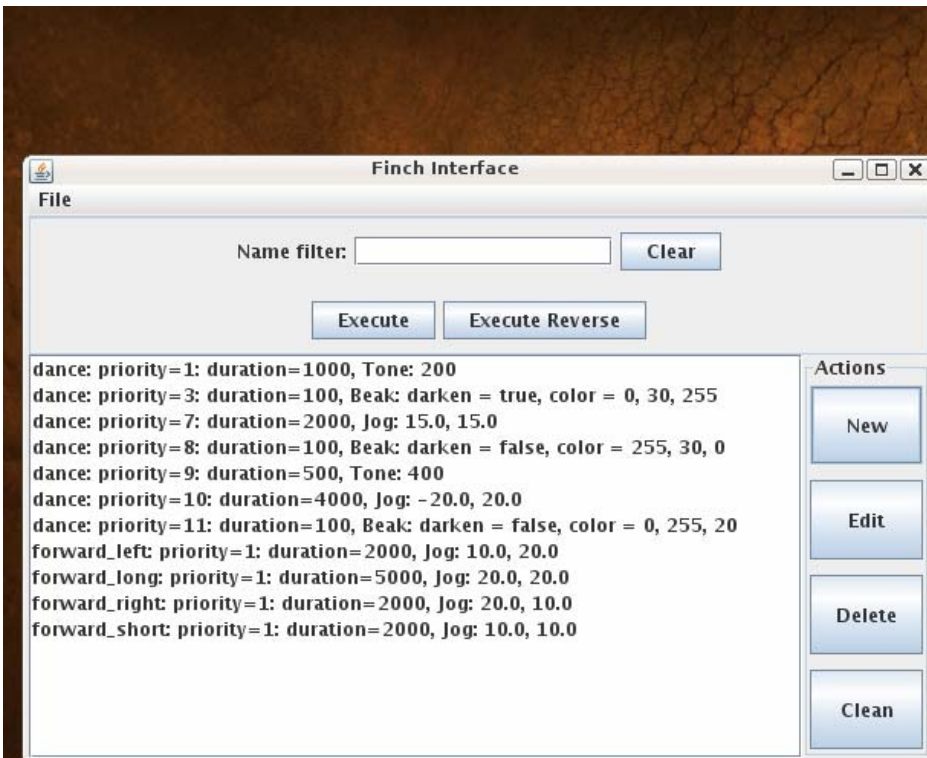
“Clean” Removes Duplicate Entries



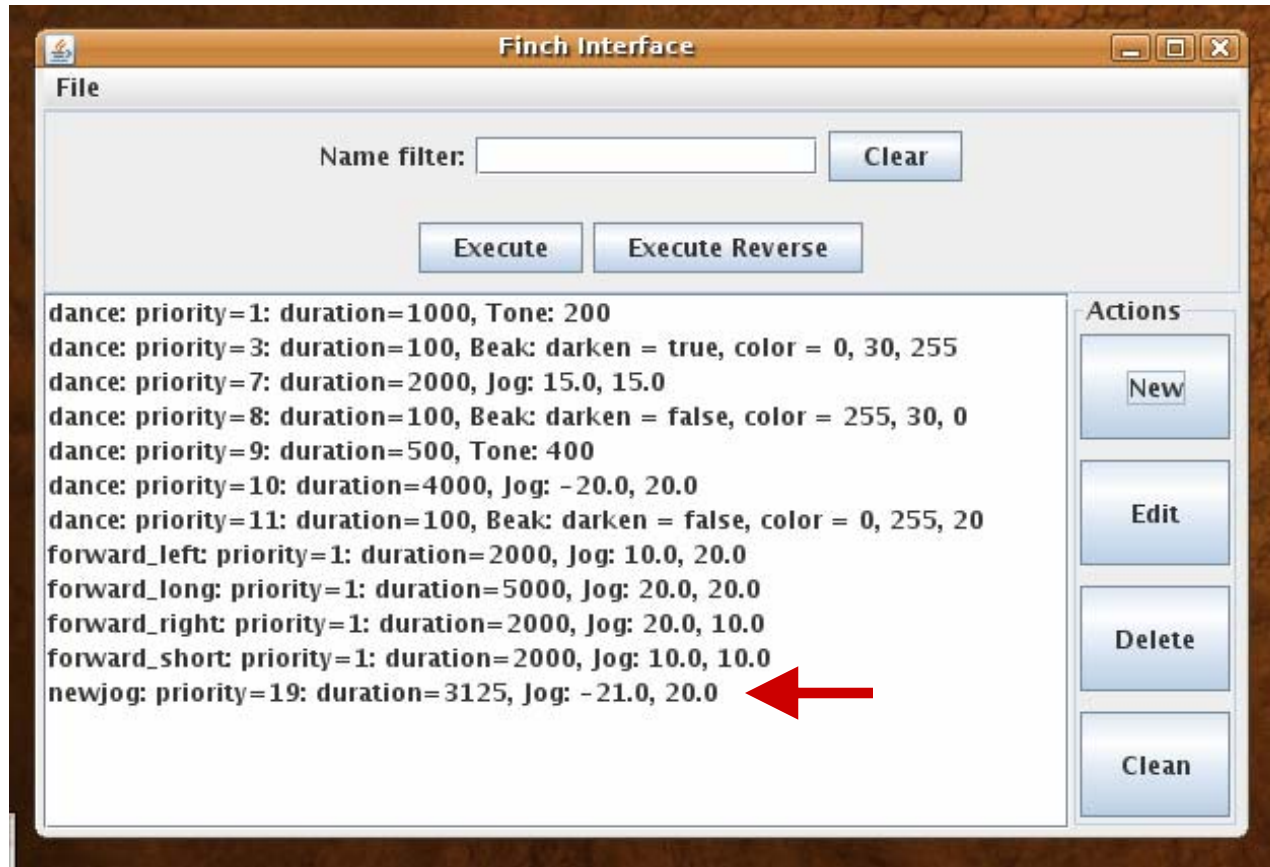
“New” Opens Dialog with Default Parameters



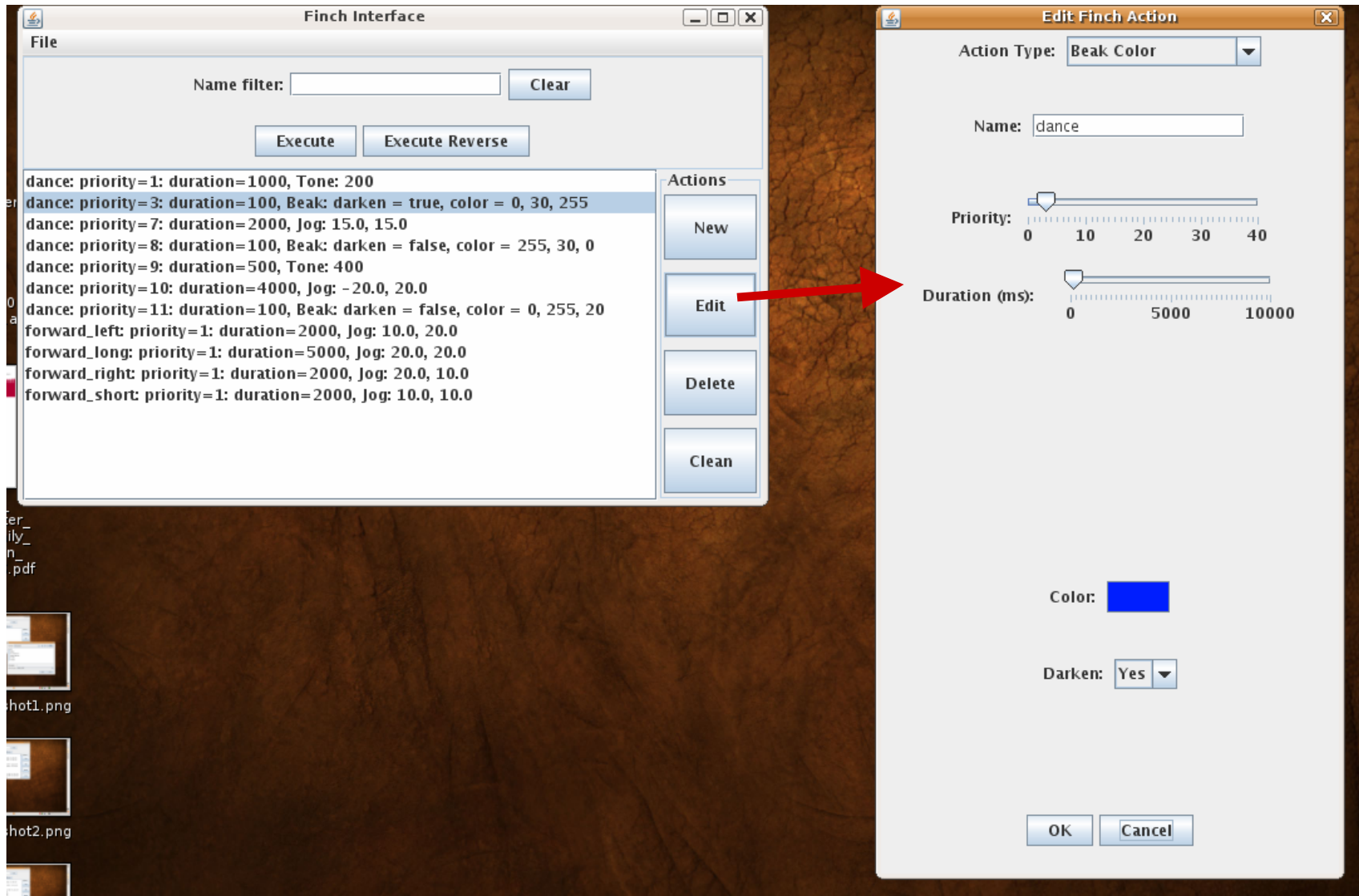
New Jog Action Specified



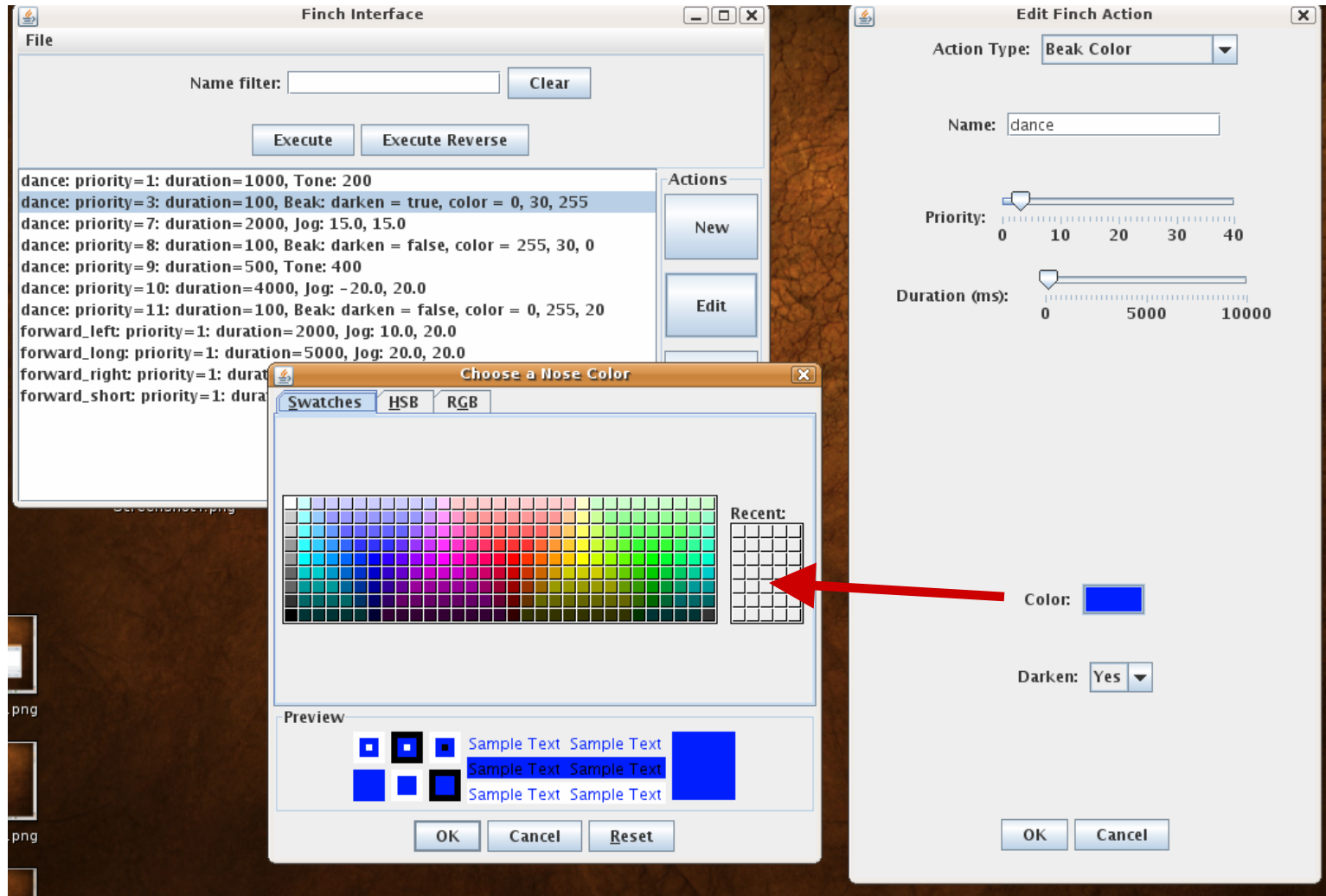
New Action Inserted into Sorted List



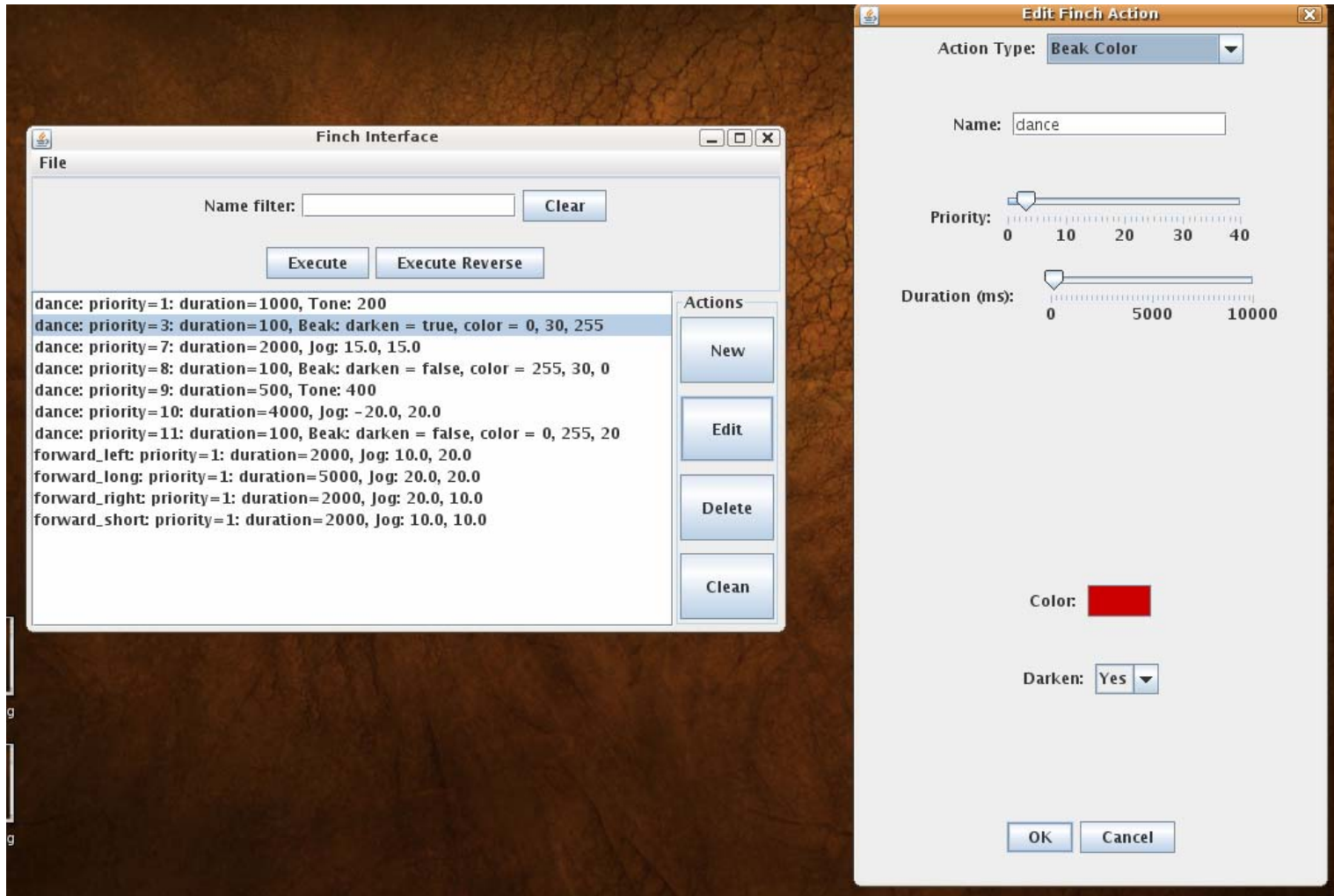
“Edit” Copies Action to Edit Dialog



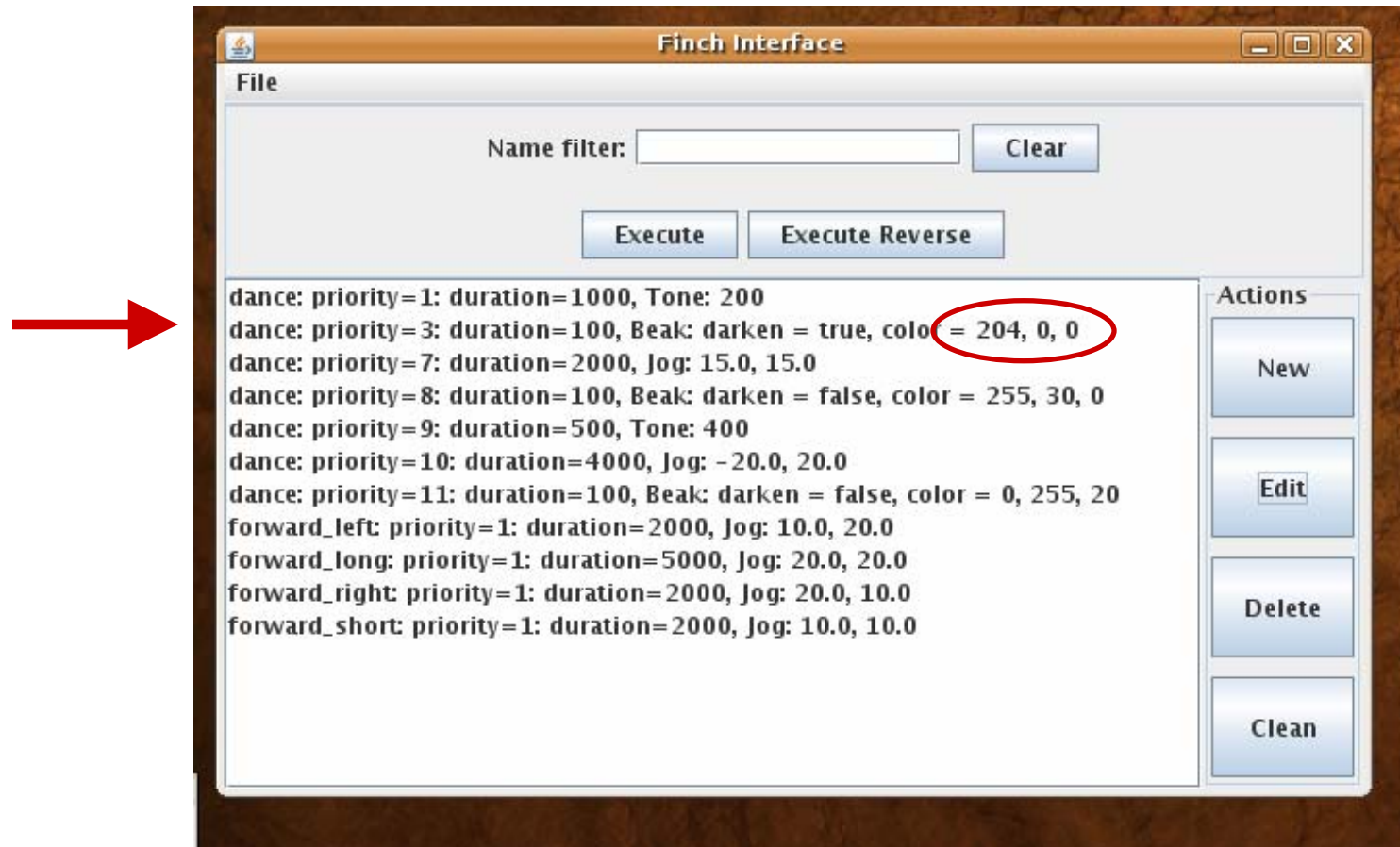
Clicking “Color” Opens Color Dialog Box



New Color Selected



“OK” Replaces Old Action with New



The Mantra...

- Implement and test incrementally
- Ask plenty of questions
- Start early