

Walking the (Mac) Responder Chain

Walking the (Mac) Responder Chain

Andy Lee
MacTech Conference 2014

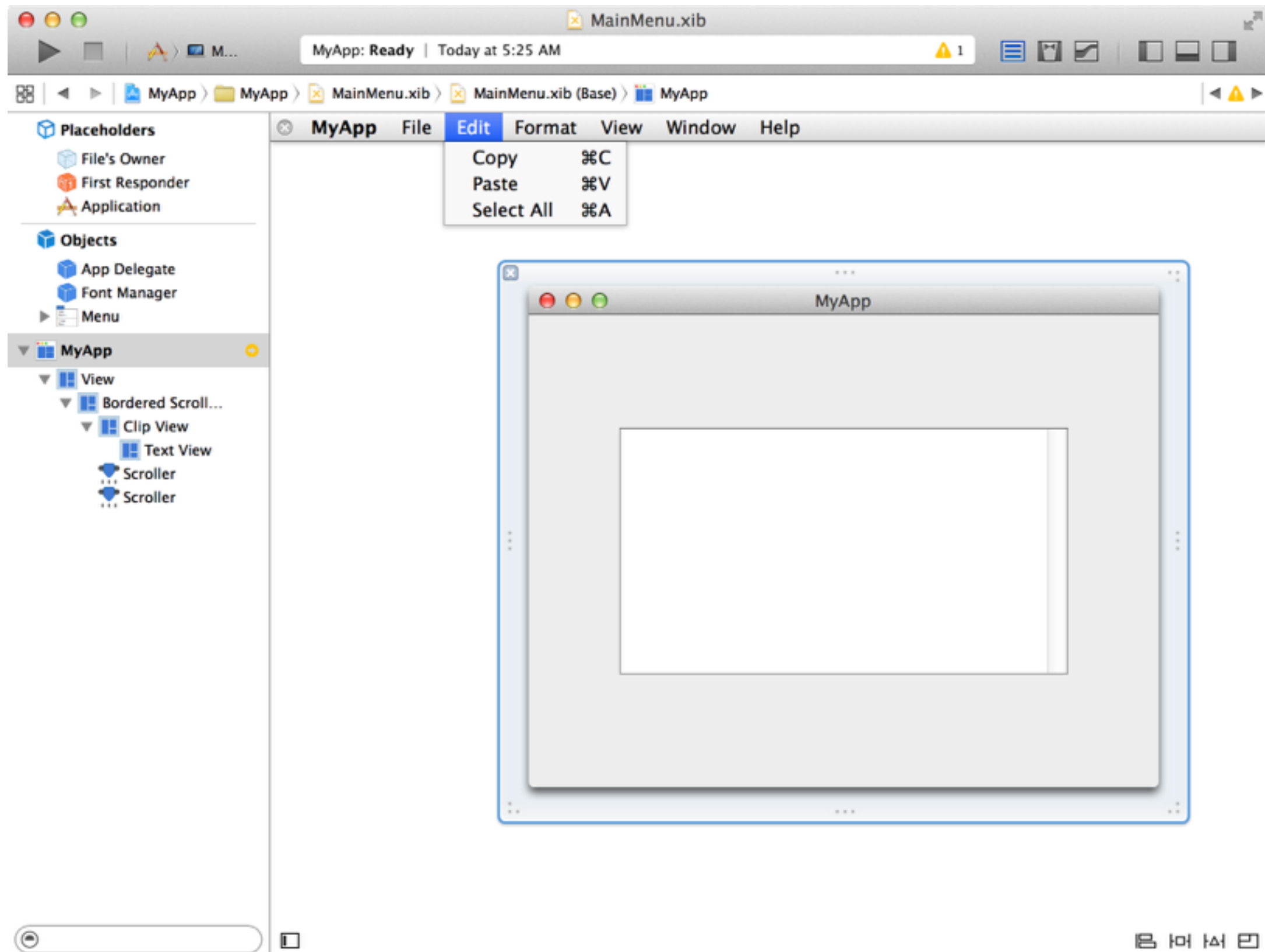
"Cocoa Application Competencies"

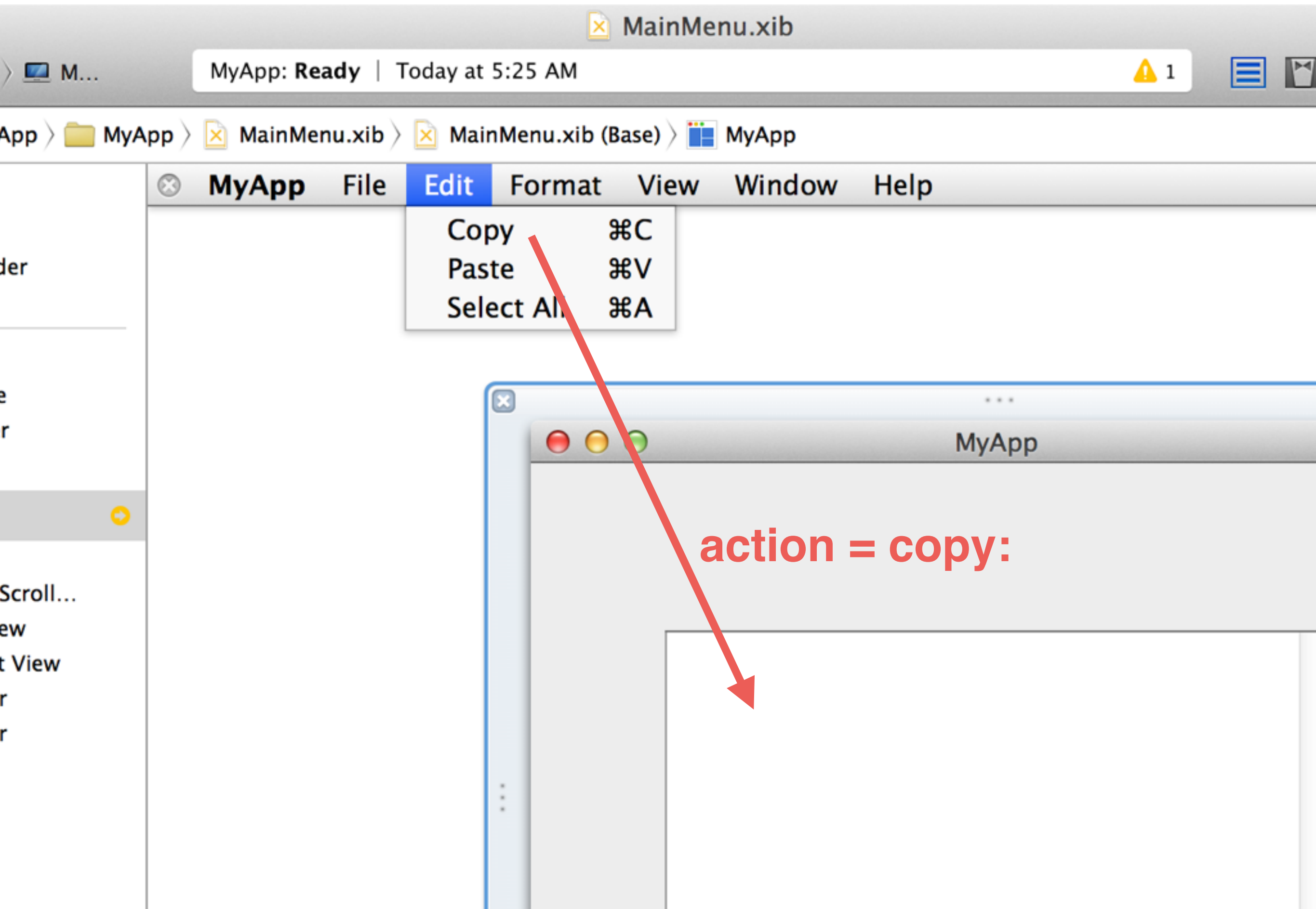
- Animation
- Application object
- Control object
- Coordinate system
- Drawing model
- Events (iOS)
- Formatter
- Main event loop
- Outlet
- Target-Action
- Undo manager
- View hierarchy
- View object
- Storyboard
- Window object
- Pasteboard
- Preferences
- Responder object

For iOS → Mac devs

- Many of the differences from iOS relate to event dispatch or the responder chain.
- Notable exception is bindings.

Conceptual beginnings:
target-action

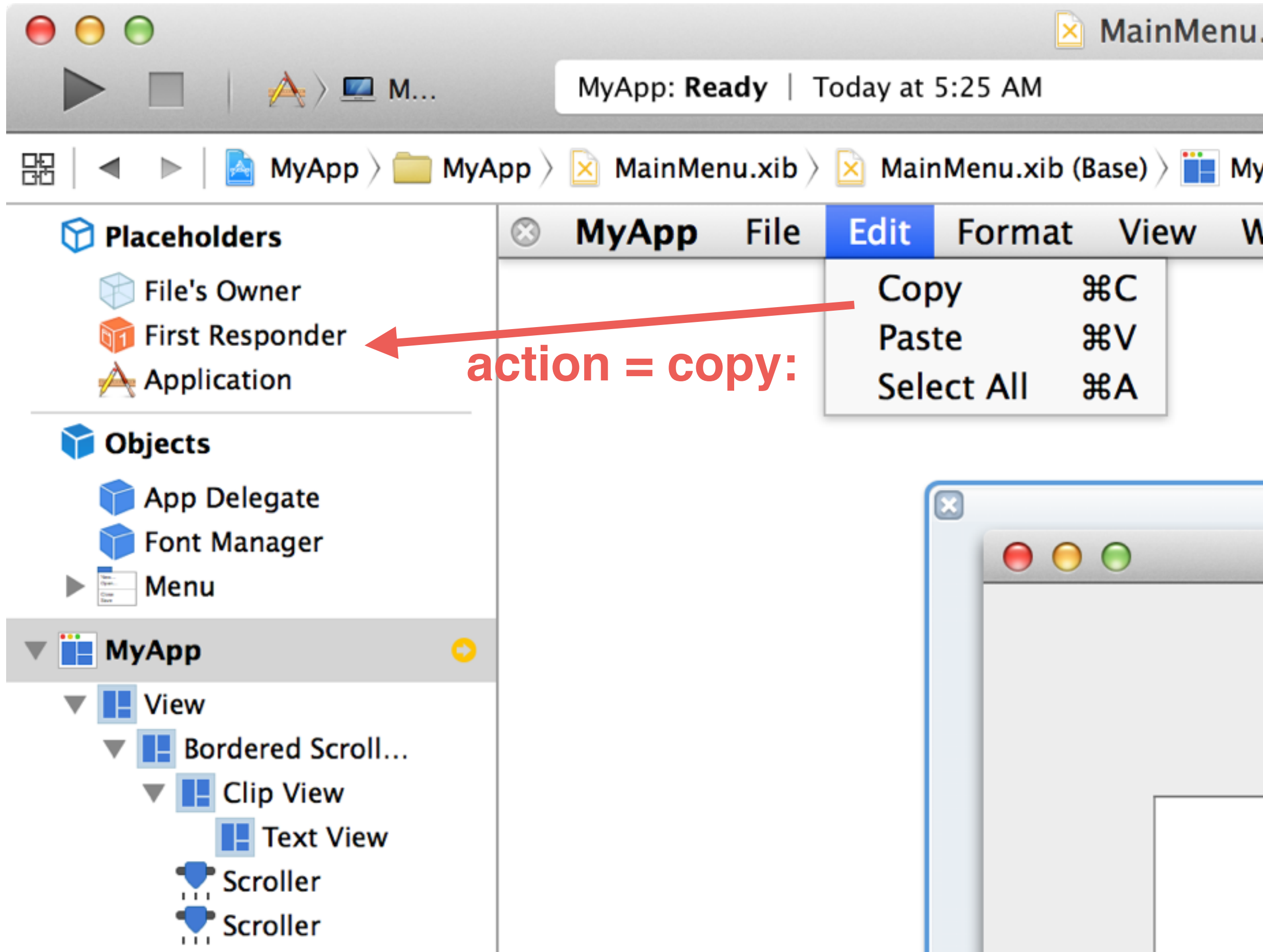




```
[copyMenuItem  
    setTarget:myTextView];  
[copyMenuItem  
    setAction:@selector(copy:)];
```


Problem!

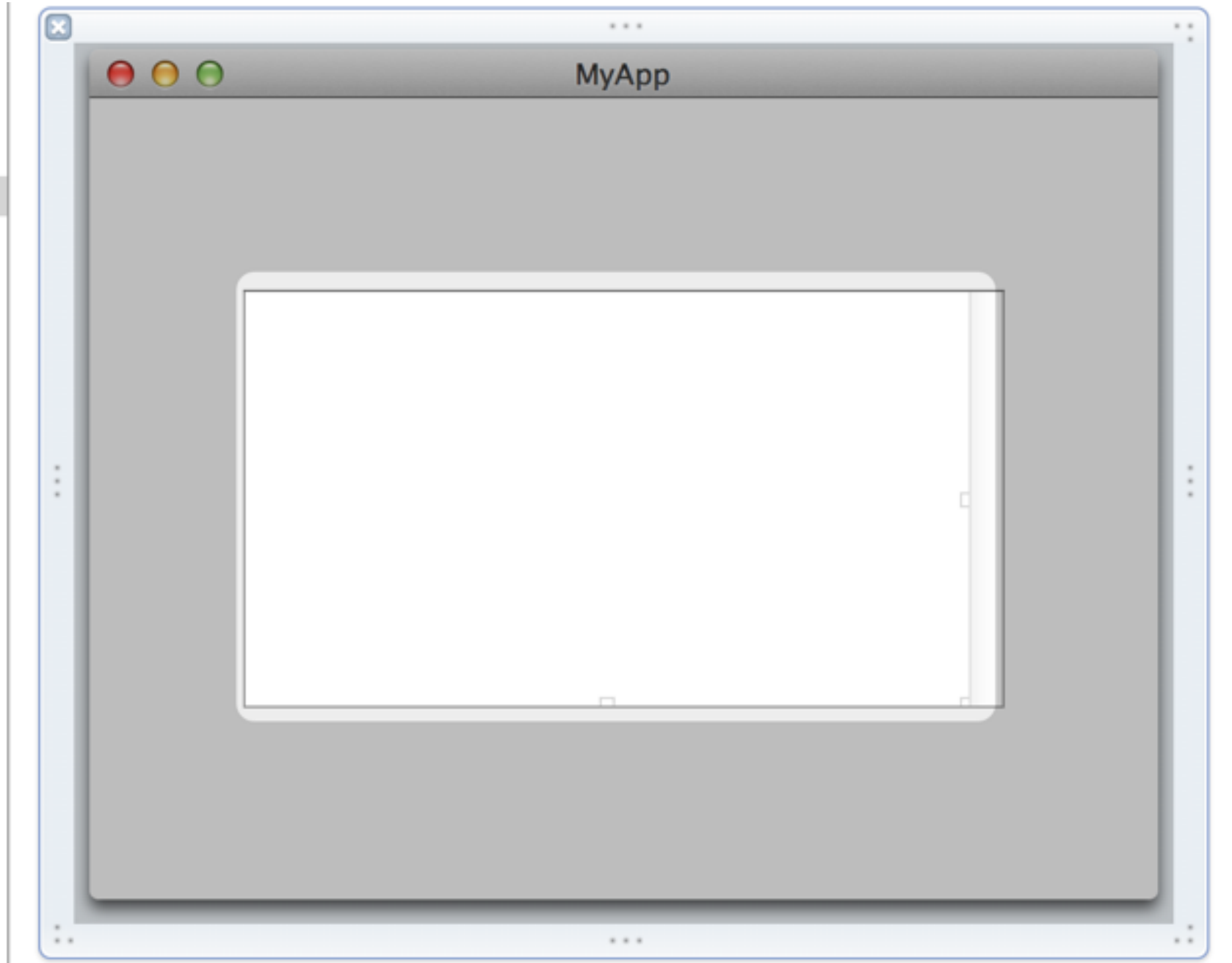
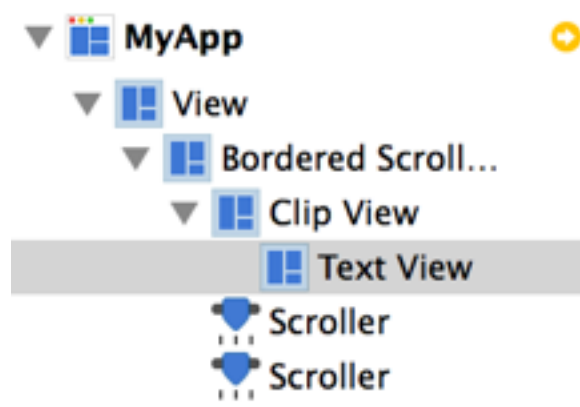
Solution:
First Responder

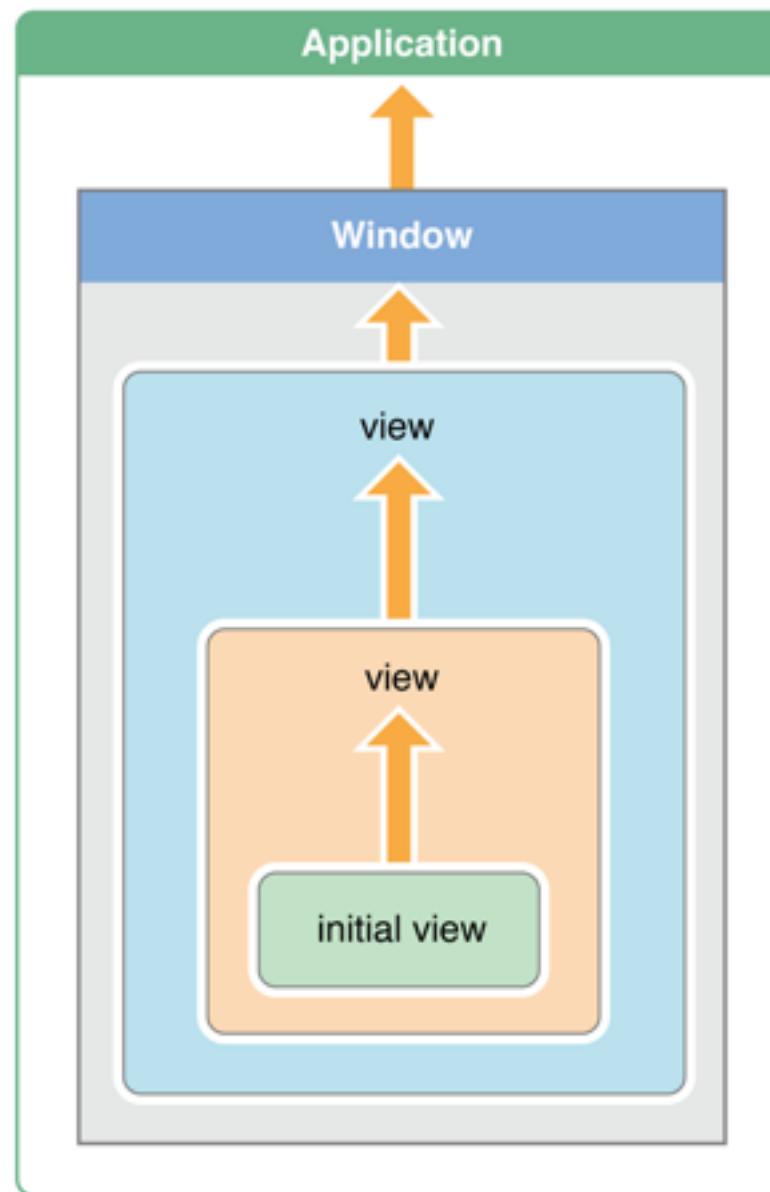


```
[copyMenuItem  
    setTarget:nil];
```

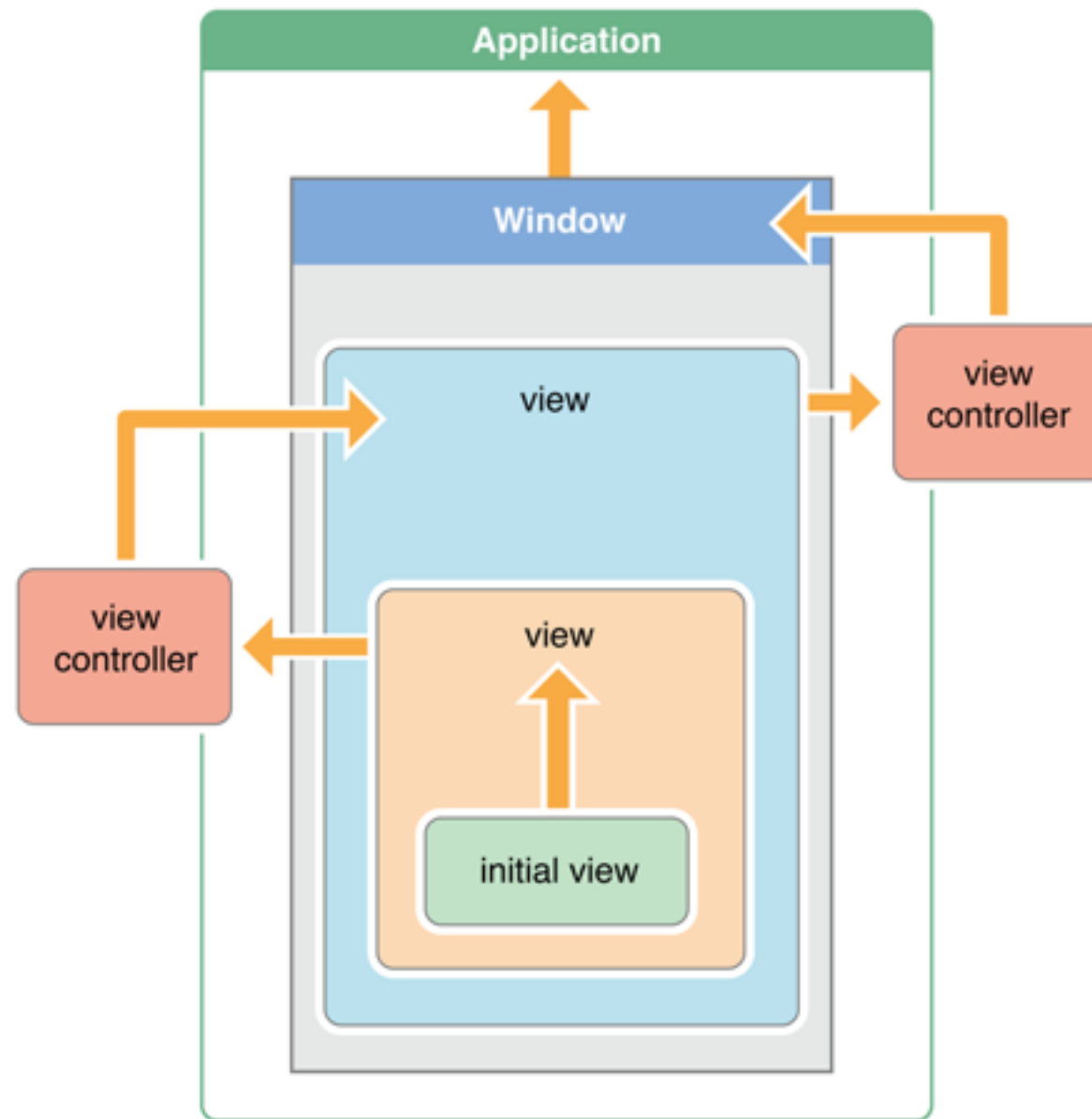
```
[copyMenuItem  
    setAction:@selector(copy:)];
```

nextResponder and
the responder chain

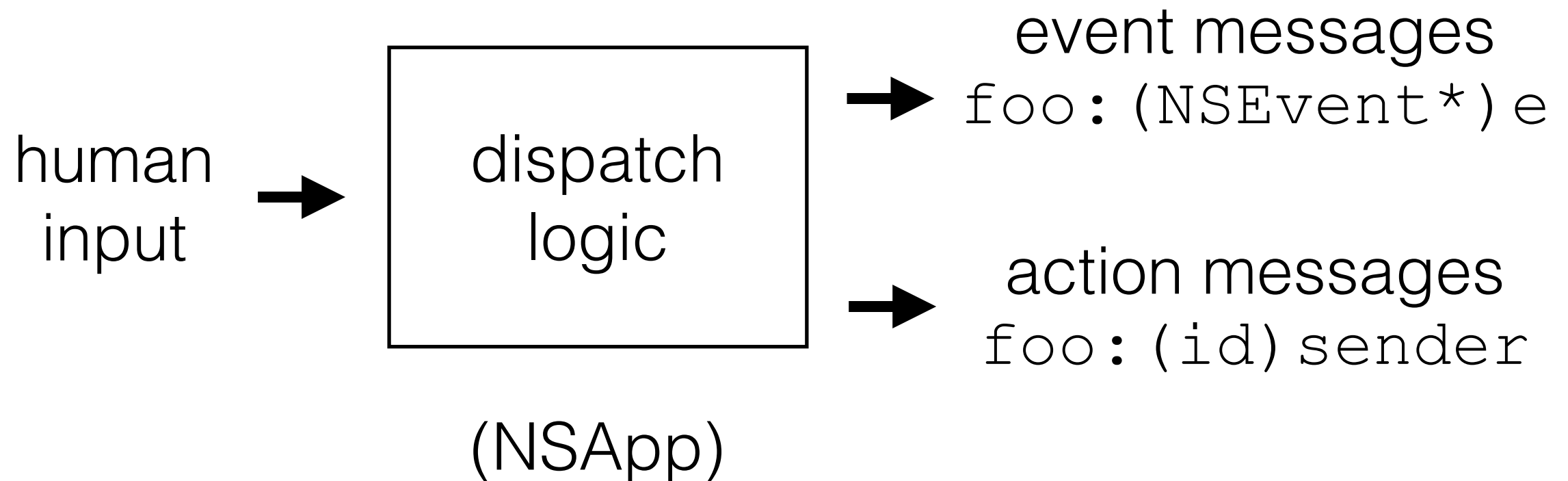




(Consider how NSScrollView gets messages.)

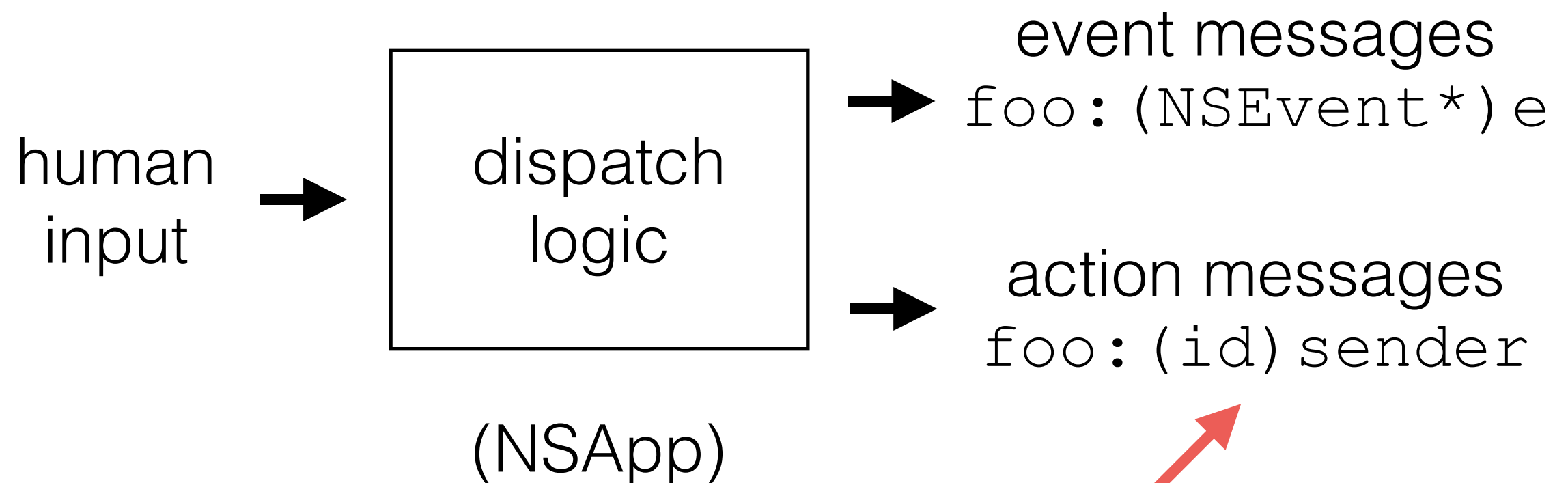


Both iOS and Mac



Dispatch logic

- Which object should receive the event or action message?
- In the case of an action message, which object should be passed as the sender?



Usually sent to **NSResponder** objects

NSResponder

- Event methods
 - `mouseDown:`
- Action methods
 - `insertNewline:`
- Chaining and unchaining
 - `nextResponder`
- First responder methods
 - `acceptsFirstResponder`

iOS/Mac differences

Mac: mouse

- Modifier keys
- Left-click, right-click, middle-click.
- Hover behavior (e.g. change cursor, highlight)

Mac: keyboard

- Keyboard is always available
- Modifier keys
- Kinds of event
 - Key equivalent (e.g. for menu, button)
 - Keyboard interface control (e.g. Tab)
 - Keyboard action (key bindings, e.g. `pageUp:`)
 - Character input

Mac: responder chain

- Responder chain is more complex.
 - E.g. can include both key and main windows
 - Different if app is document-based.
- Used for additional things:
 - `NSUserInterfaceValidations` (enabling/disabling)
 - Menu items
 - Toolbars
 - Buttons

- BTW, responder chain is also used for:
 - Undo
 - NSError handling (`presentError:` methods)

To ponder: target-action
sends no context except
sender.

- Sometimes target's internal state provides context.
- Sometimes target can get info from global object, e.g. pasteboard.
- Sometimes integer suffices as context, can use `sender.tag`.
- Some senders (e.g. `NSMenuItem`) have a `representedObject` that can contain context.
- Some tricky cases e.g. view-based table.

See also

- Apple:
 - "Cocoa Event Handling Guide"
 - "Cocoa Application Competencies"
- *Cocoa Design Patterns*, Buck & Yacktman, 2009 (!)
 - Only quibble: "down" vs. "up"
- Blog posts by Brent Simmons and Guy English
 - Google either name plus "responder chain".
- Google "Modernizing the Responder Chain" -- interesting ideas by Tim Wood.

@cocoadoog

CocoaHeadsNYC.org

NotesFromAndy.com