CS193E
Lecture 10

Events and Views
Selection

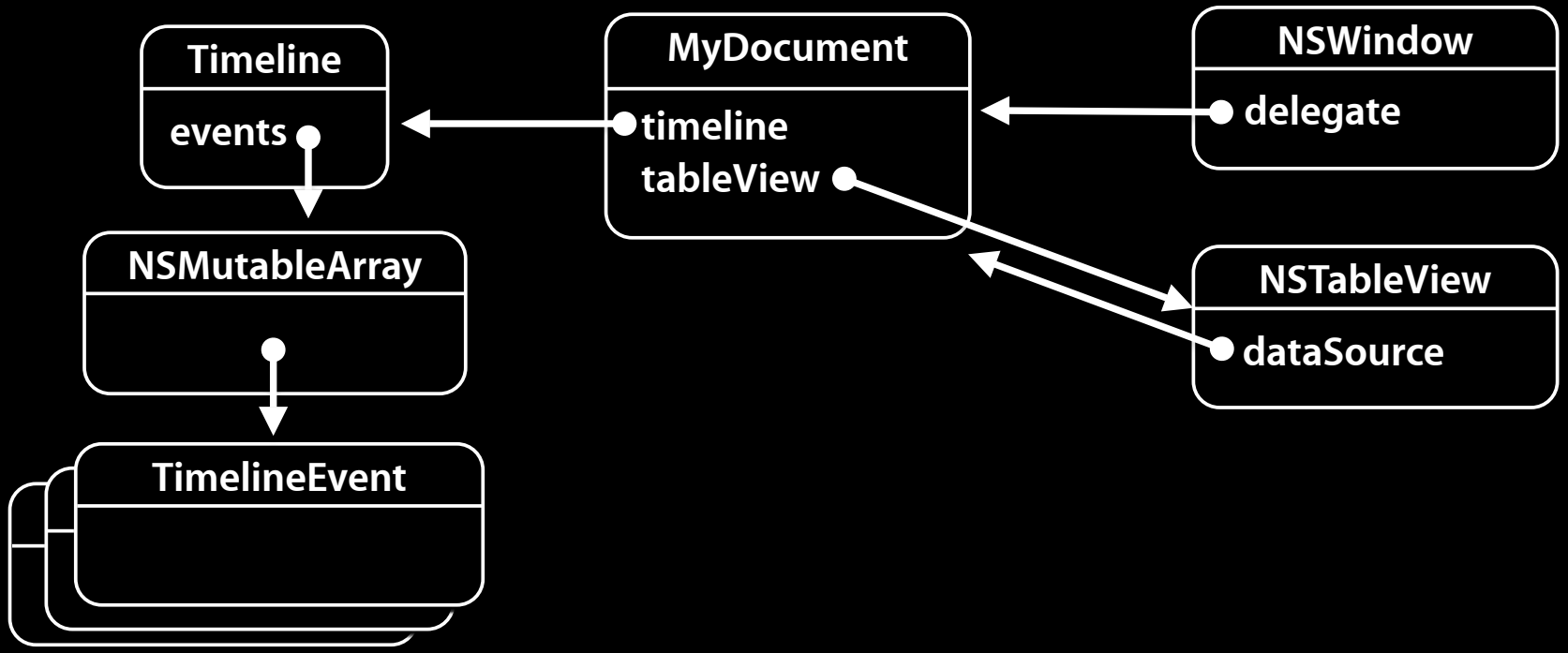
Today's Topics

- Questions?
- Personal Timeline II
- Event Model
- Views & Events
- Selection
- Miscellaneous

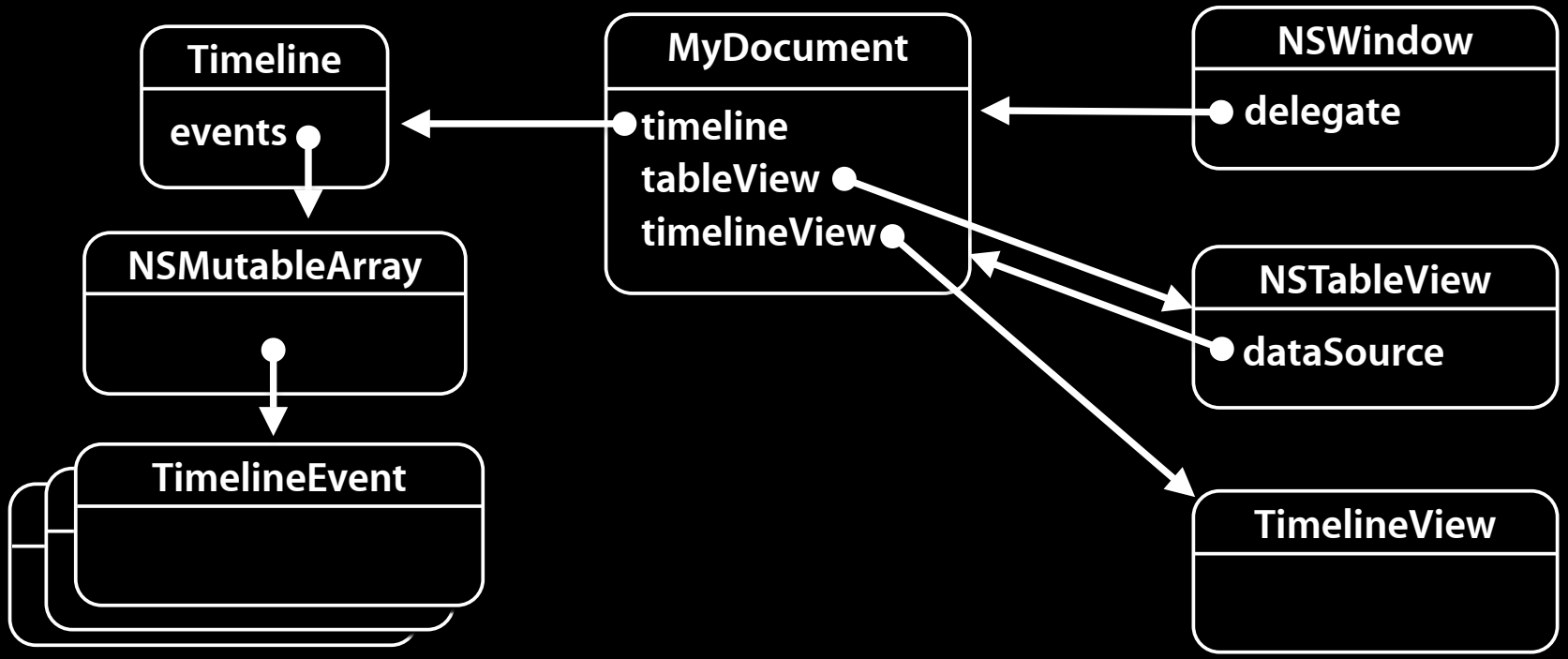
Demo

Personal Timeline II

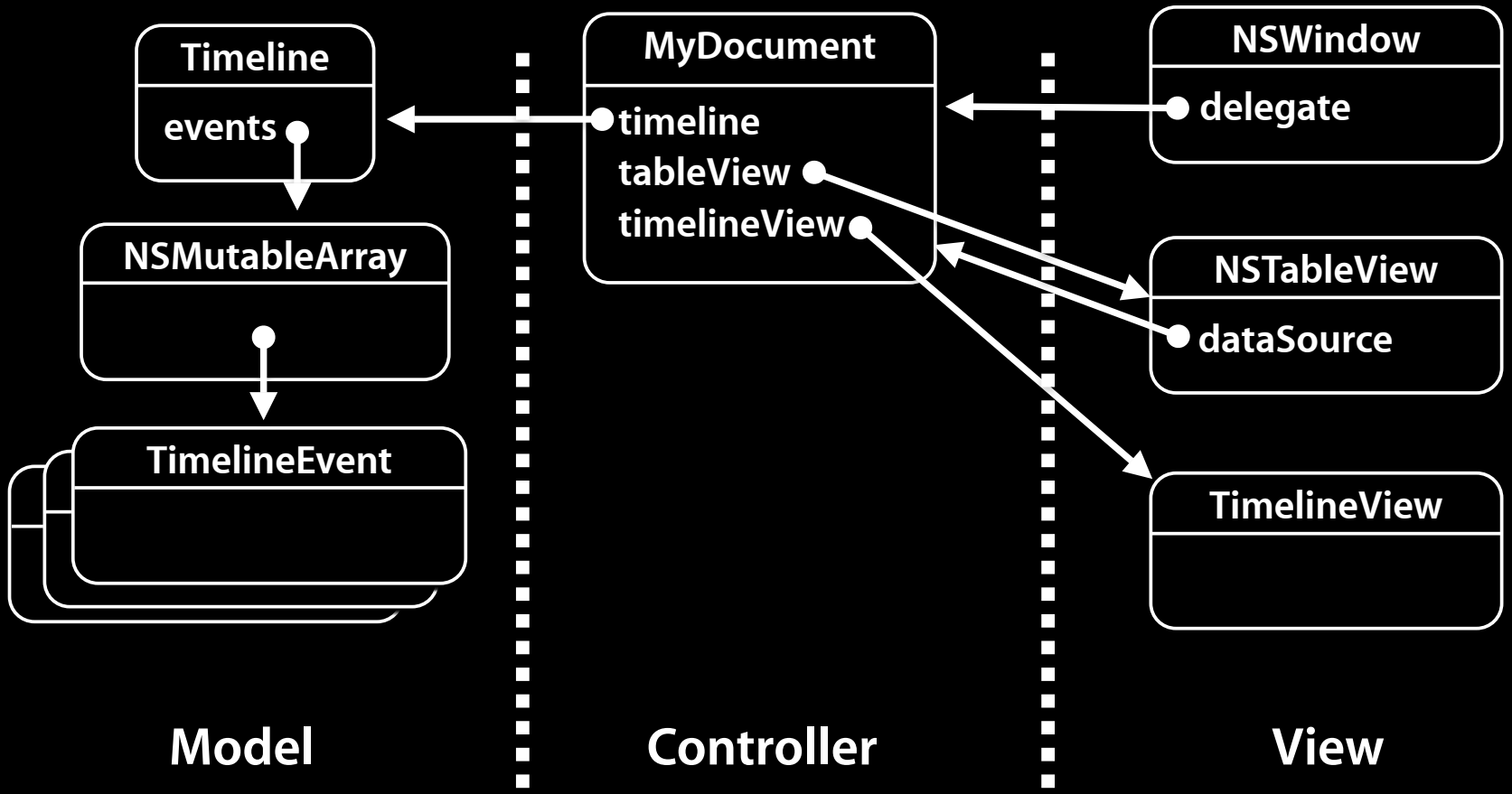
Personal Timeline I



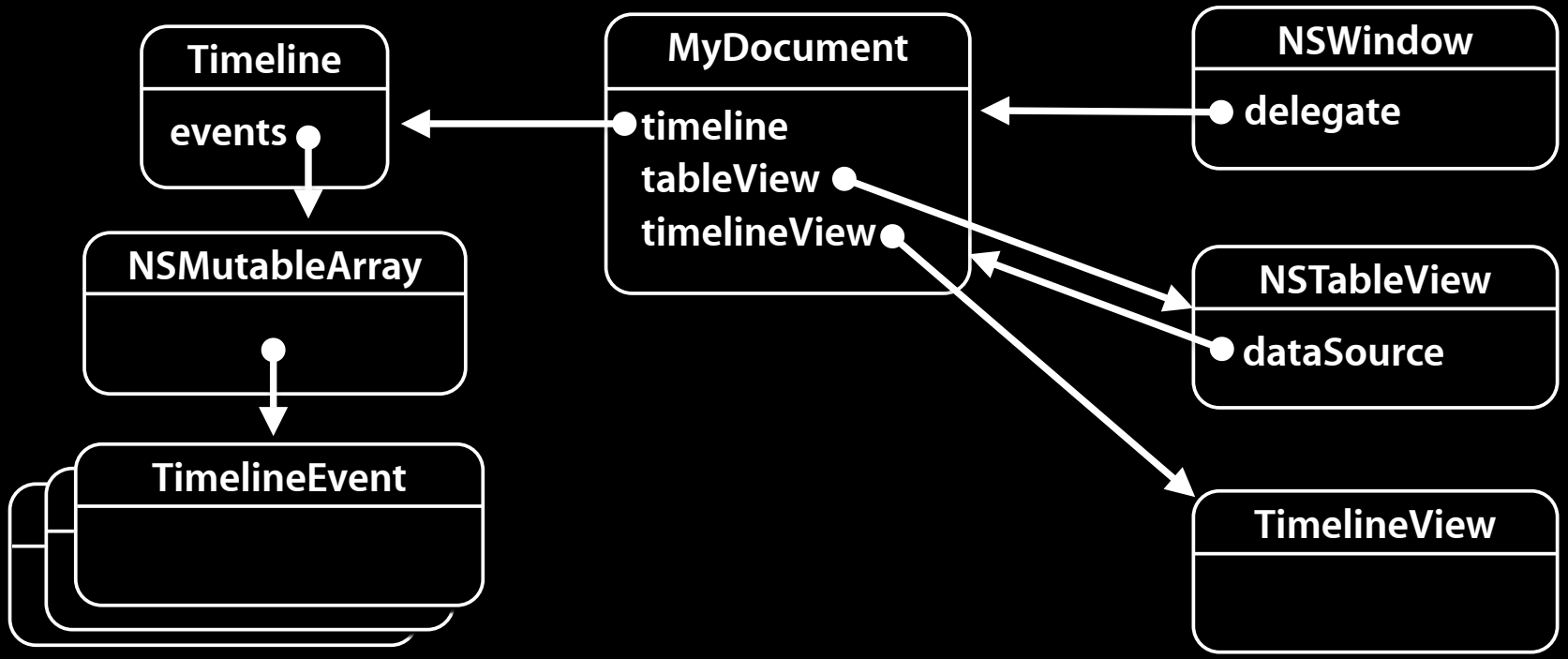
Personal Timeline II



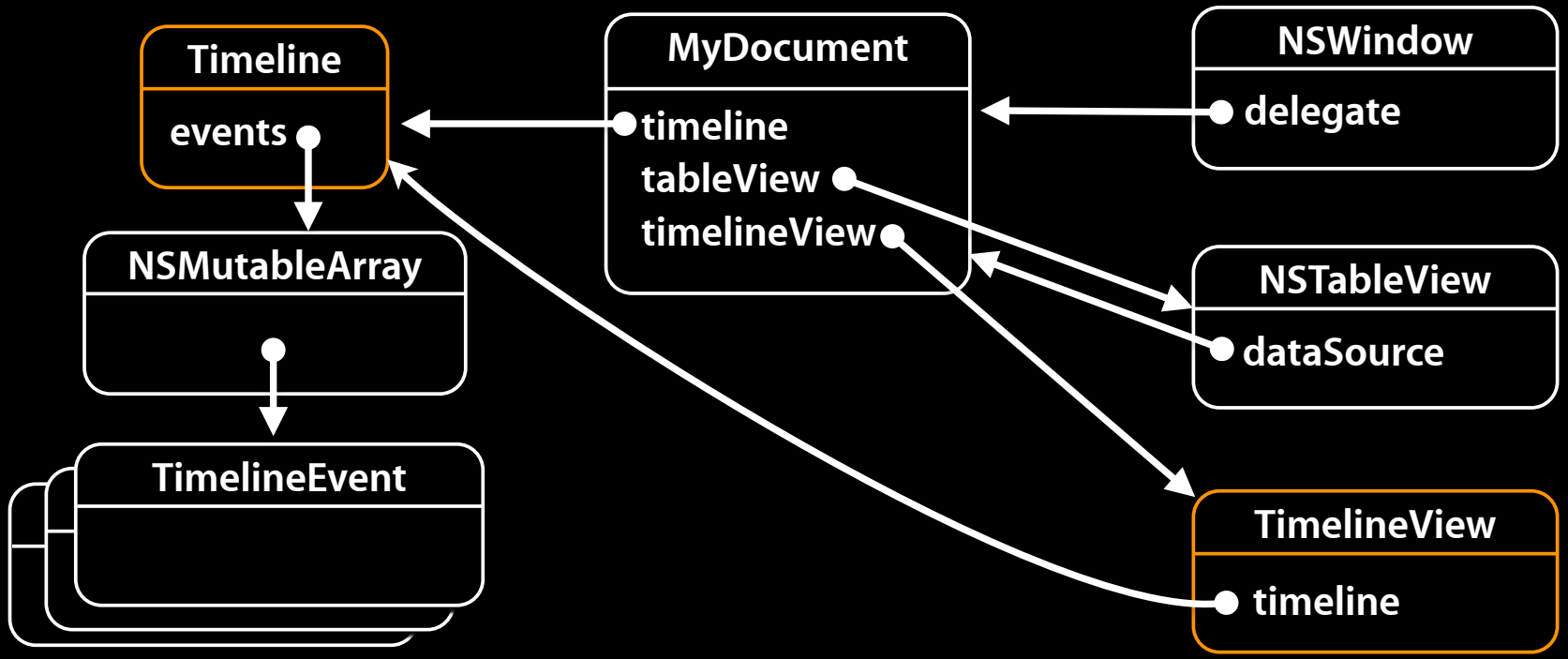
Personal Timeline II



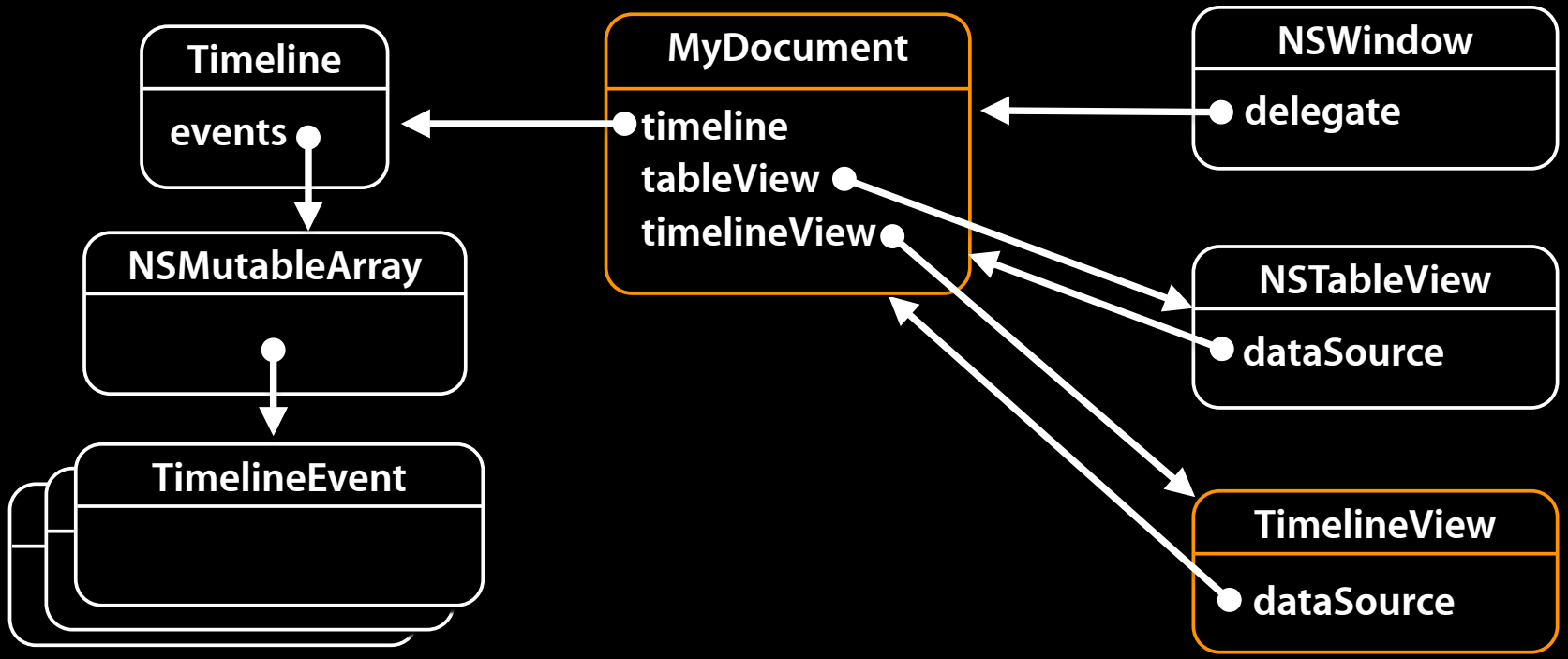
How to get data into timeline view?



How to get data into timeline view?

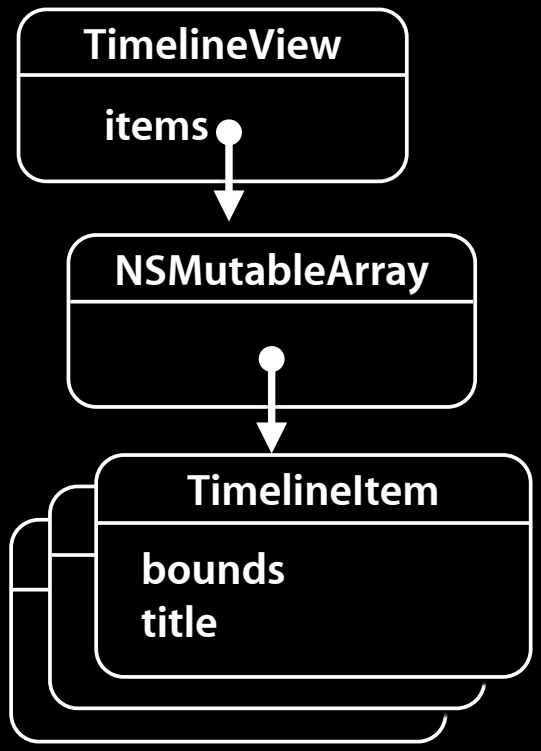


How to get data into timeline view?



TimelineView

Each item encapsulates drawing



Event Model

main

```
 int main(int argc, char **argv)
{
    return NSApplicationMain(argc,argv);
}
```

NSApplicationMain

Psuedocode of application startup

```
int
NSApplicationMain(int argc,char **argv)
{
    [NSApplication sharedApplication];
    [NSBundle loadNibNamed:@"MainMenu"
        owner:NSApp];
    [NSApp run];
    return 0;
}
```

[NSApp run]

Psuedocode of a run loop

```
 - (void)run
{
    while (1) {
        /* Get next event */
        ...
        /* Dispatch event */
        ...
    }
}
```

What is an “Event”?

- Typically, some form of user input
  - From the mouse
  - From the keyboard
- Timers
  - Periodic
  - One-shot

NSEvent methods

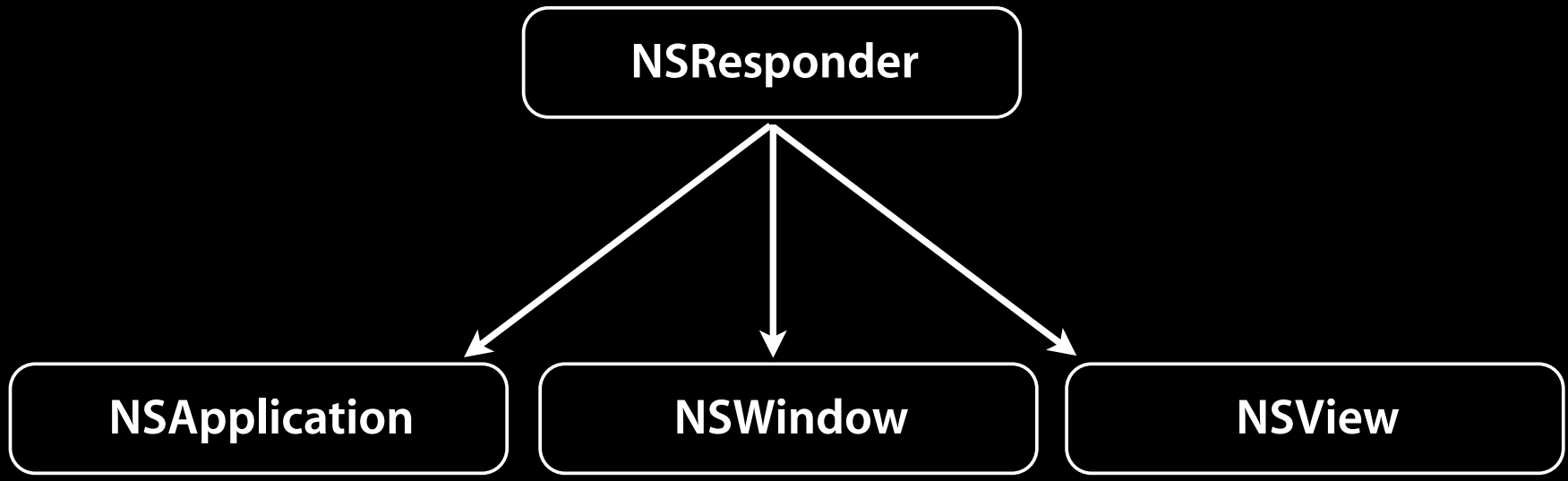
- (NSEventType)type
- (NSPoint)locationInWindow
- (unsigned int)modifierFlags
- (int)clickCount /* for mouse events */
- (NSString *)characters /* for key events */
- (BOOL)isARepeat /* for key events */

NSEventType

```
NSLeftMouseDown
NSLeftMouseUp
NSRightMouseDown
NSRightMouseUp
NSMouseMoved
NSLeftMouseDragged
NSRightMouseDragged
NSMouseEntered
NSMouseExited
NSKeyDown
NSKeyUp
NSFlagsChanged
```

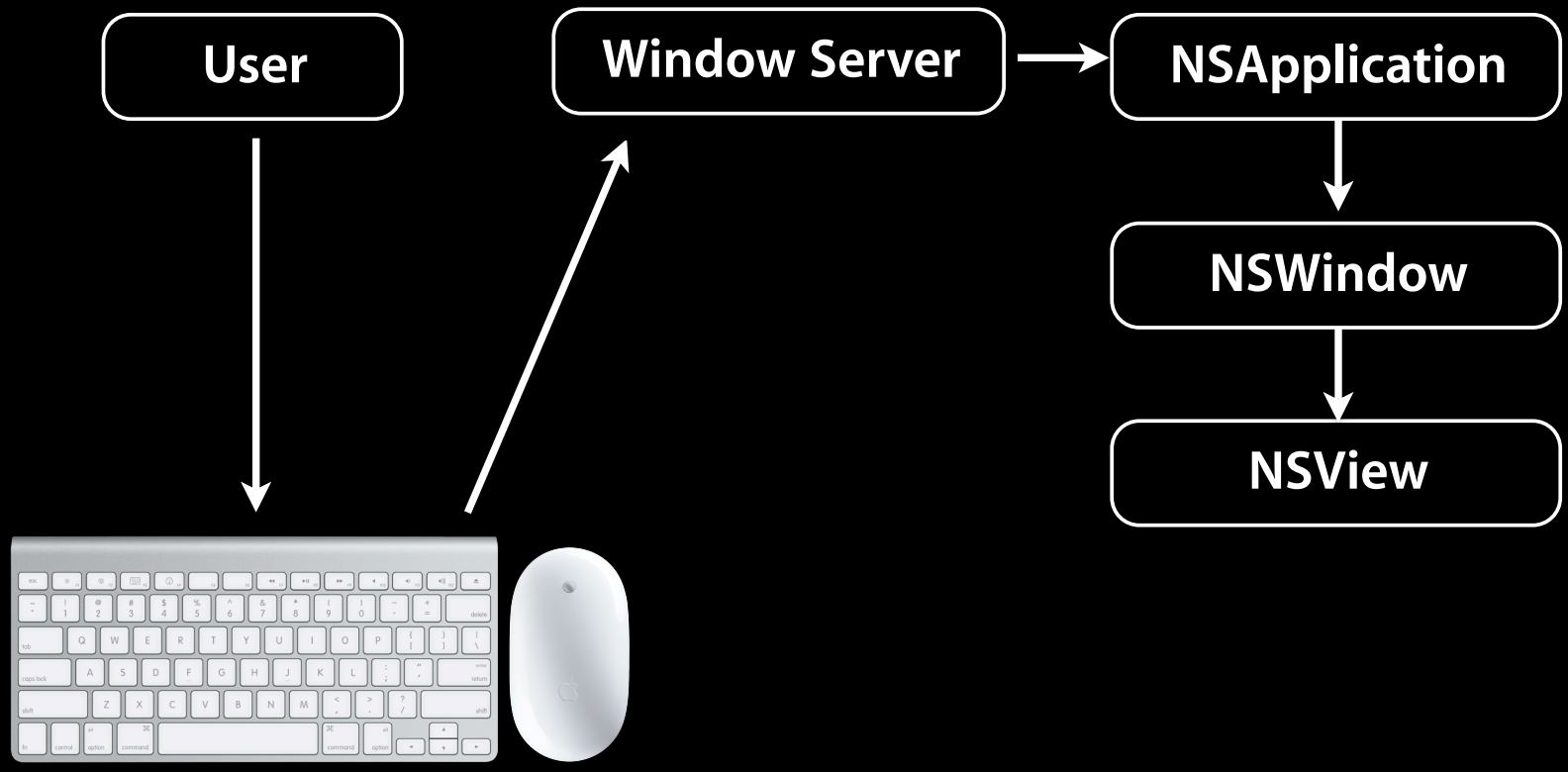
and so on

Who Processes Events?



NSResponder subclasses

Event Flow



Views & Events

Mouse Events

- The window server determines which application and window received the event.
- The application gets the event, and hands it to the target window
  - - (void)sendEvent:(NSEvent *)event
- Window performs hit testing on its contentView

Mouse Events

- The target view is sent the appropriate method
  - - (void)mouseDown:(NSEvent *)event;
  - - (void)mouseDragged:(NSEvent *)event;
  - - (void)mouseUp:(NSEvent *)event;
  - - (void)mouseEntered:(NSEvent *)event;
  - - (void)mouseExited:(NSEvent *)event;
  - - (void)mouseMoved:(NSEvent *)event;

Key Events

- Dispatched similarly to mouse events, except —which window/view receives the event?
- We don’t have a mouse position for a key event

Key Events

- NSApplication keeps track of the current keyWindow
- Each window has a view that is its firstResponder
- In IB, "First Responder" is a special target which always points to the currently "active" view
- We saw the first responder when we talked about actions and the responder chain

No-Frills Event Handling

- Implement the following:
  - - (void)mouseDown:(NSEvent *)event;
  - - (void)mouseDragged:(NSEvent *)event;
  - - (void)mouseUp:(NSEvent *)event;

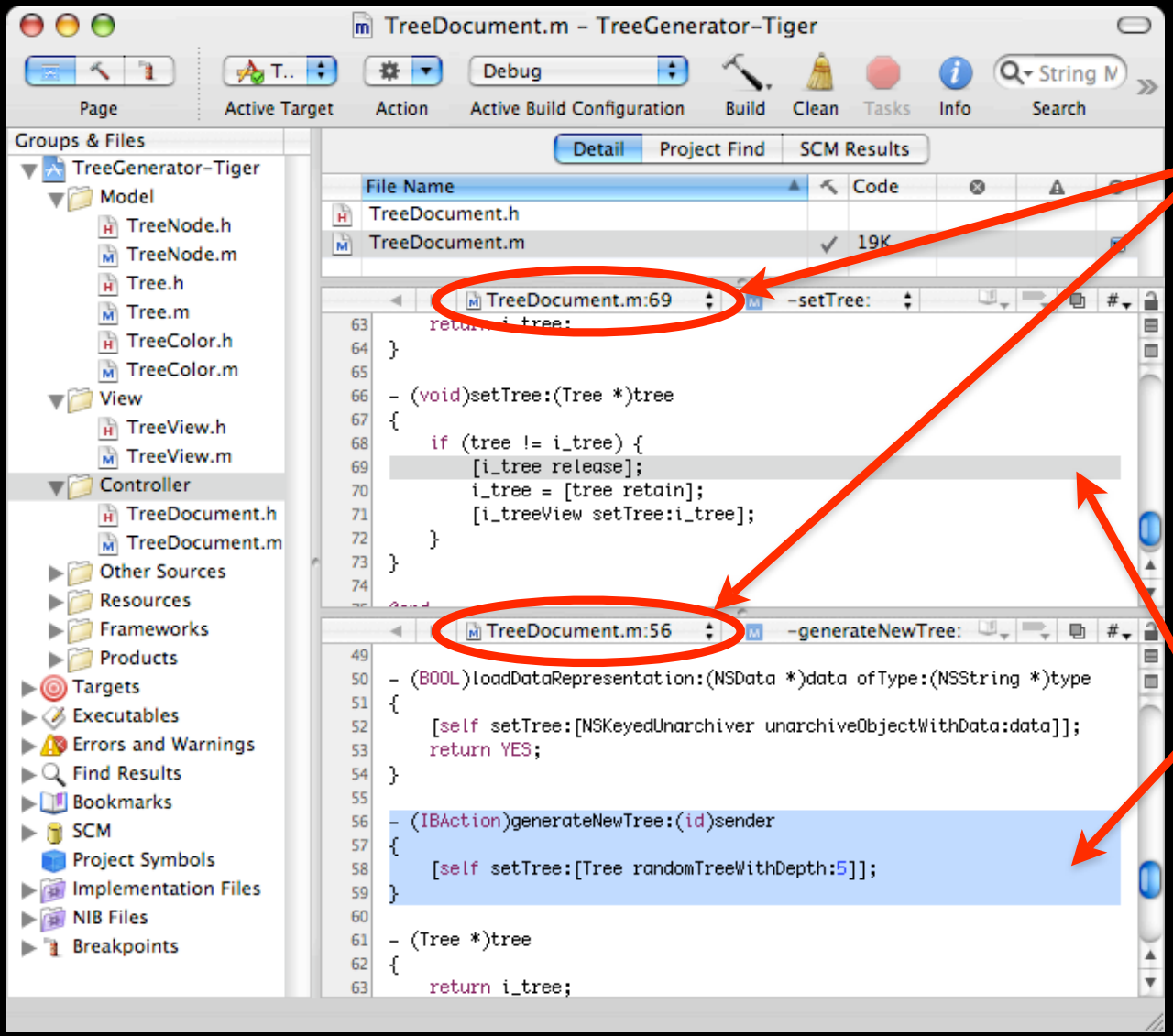
Demo

Very basic event handling

Selection Handling

Who Owns Selection?

- We've got Models, Views and Controllers — which one should keep track of selection?
- Depends on the behavior you're looking for!
- If you've got multiple views on your model, do you want selection to be synchronized?
- Typically you don't, but there are times when you do

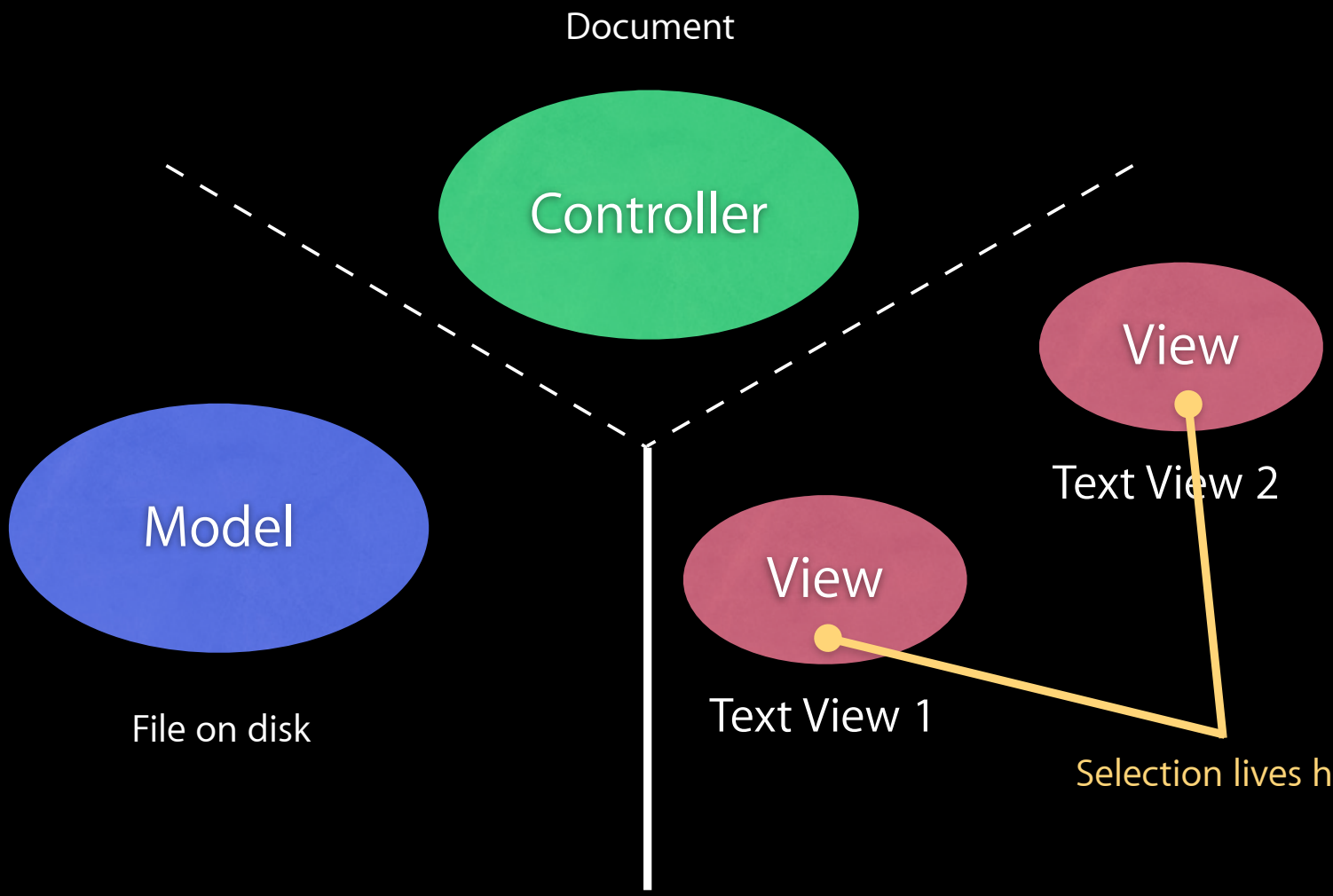


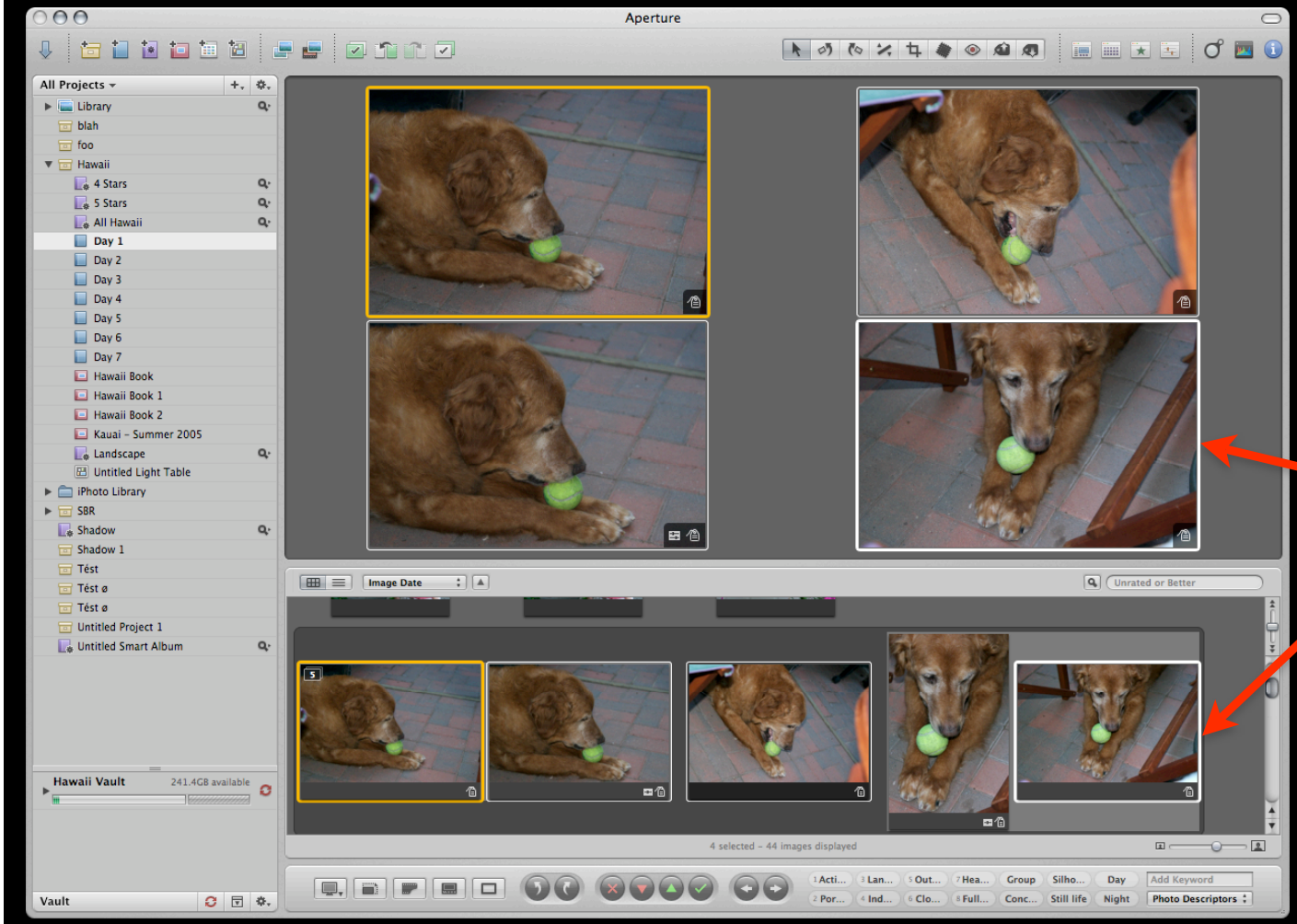
Same Document

Same Model
(underlying file)

Multiple Views
with separate
selection

Xcode Selection Example





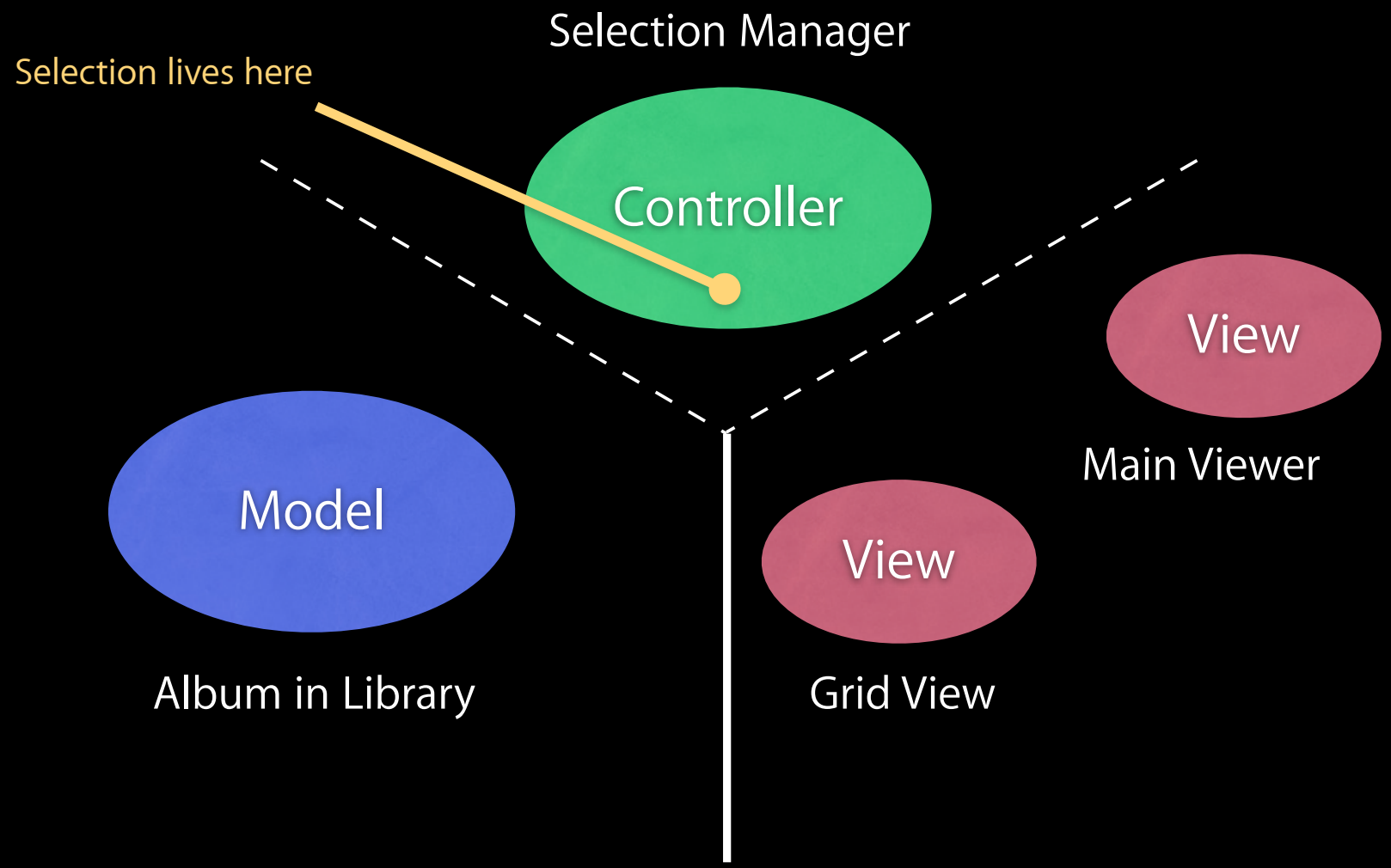
Same Document (album in library)

Same Model (picture of Shadow)

Multiple Views

Selection stays in sync even in different views

Aperture Selection Example



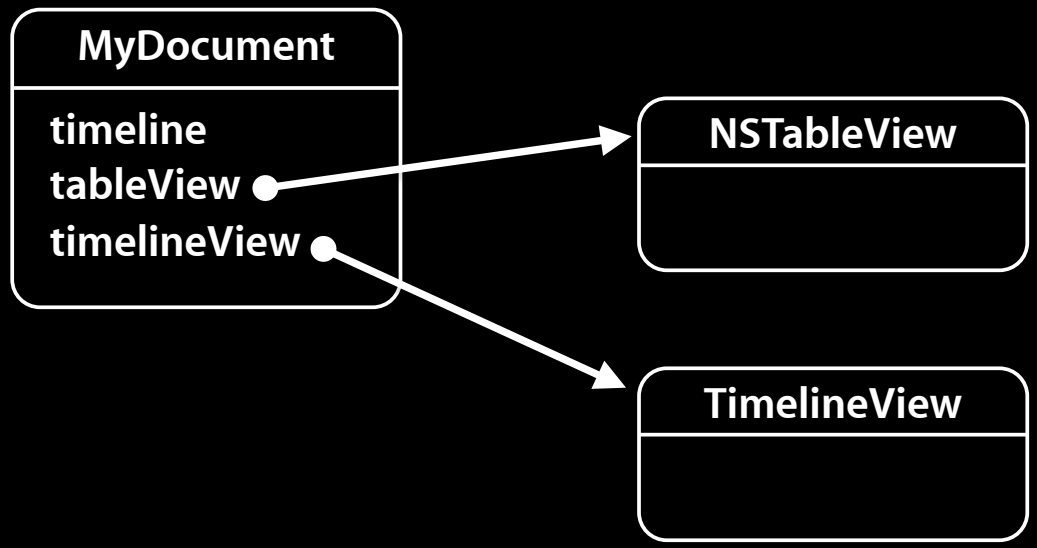
Designing Selection

- Decide on behavior you need, especially in context of multiple views
- Map that to the classes in your MVC design

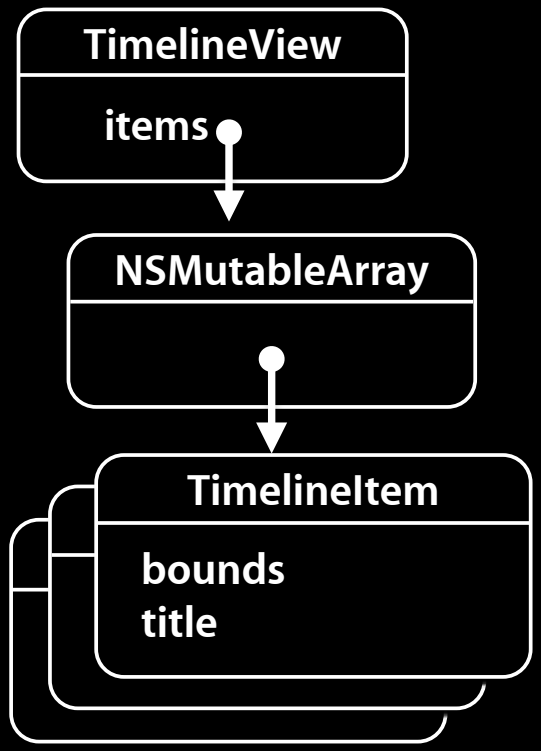
Saving Selection State

- Another question that commonly comes up is whether to save selection state?
- Some apps do, some apps don't (thankfully more and more are)
- Adds a level of polish to give that "restore the user's state to what they were doing before"
- Examples: Xcode, Mail, Keynote, Preview, iTunes (kind of)
- Finally, should saved selection state be part of the document, or stashed away in user preferences?

Selection


MyDocument
timeline
tableView
timelineView
NSTableView
TimelineView

Drawing ‘selection’



Miscellaneous

Index sets and table views
A little more about drawing

NSIndexSet

- Class holds a unique set of non-negative integers
- Used by NSTableView to represent the indexes of selected rows
- Has mutable subclass NSMutableIndexSet

Setting table view selection

```
unsigned int rowToSelect = 3;

NSIndexSet *indexSet =
      [NSIndexSet indexSetWithIndex:rowToSelect];

[tableView selectRowIndexes:indexSet
                       byExtendingSelection:NO];
```

- Use an empty set to deselect all rows

```
[NSIndexSet indexSet]; // empty set
```

View Updating

Updating the Display

- All views implement
  - (void)drawRect:(NSRect)rect;
- Never call drawRect: directly!
  - If you do, you'll get remarkably incorrect results

Updating the Display

- When the entire view needs to be redrawn, tell it:

```
[view setNeedsDisplay:YES];
```

- Through a historical quirk, setNeedsDisplay: takes an unnecessary boolean argument

Selective Update

- When only part of a view needs to be redrawn, narrow the area for redisplay:

```
[view setNeedsDisplayInRect:dirtyRect];
```

- setNeedsDisplay: is essentially equivalent to

```
[view setNeedsDisplayInRect:[view bounds]];
```

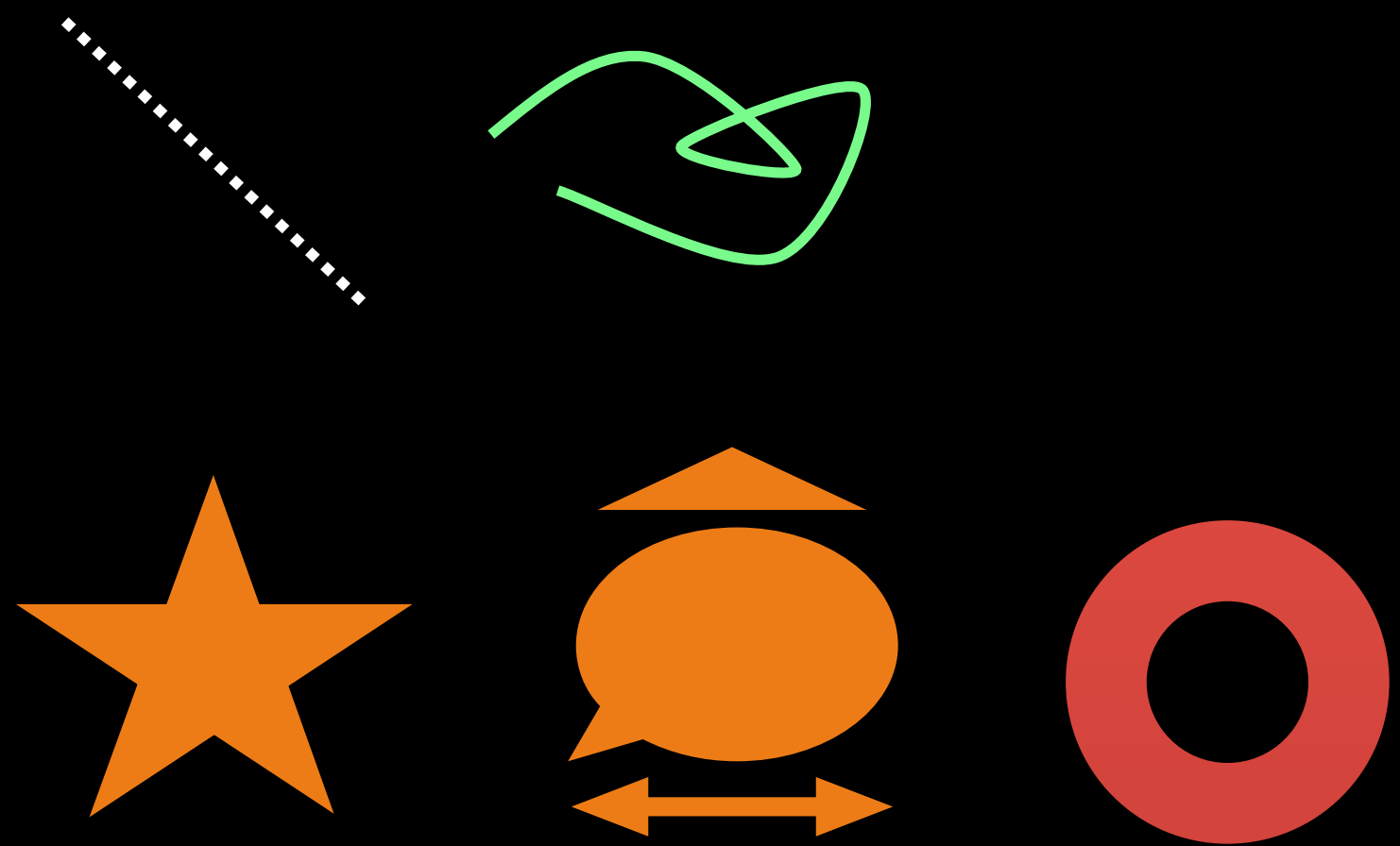
- Dirty rects are batched up and redrawn at the end of the event loop, before the next event is processed

Path Creation

NSBezierPath

- Follows the PostScript/PDF drawing model
  - Paths are sequences of straight or curved line segments
  - A single path can be closed or open
  - A path can consist of several disconnected subpaths

Examples of Paths

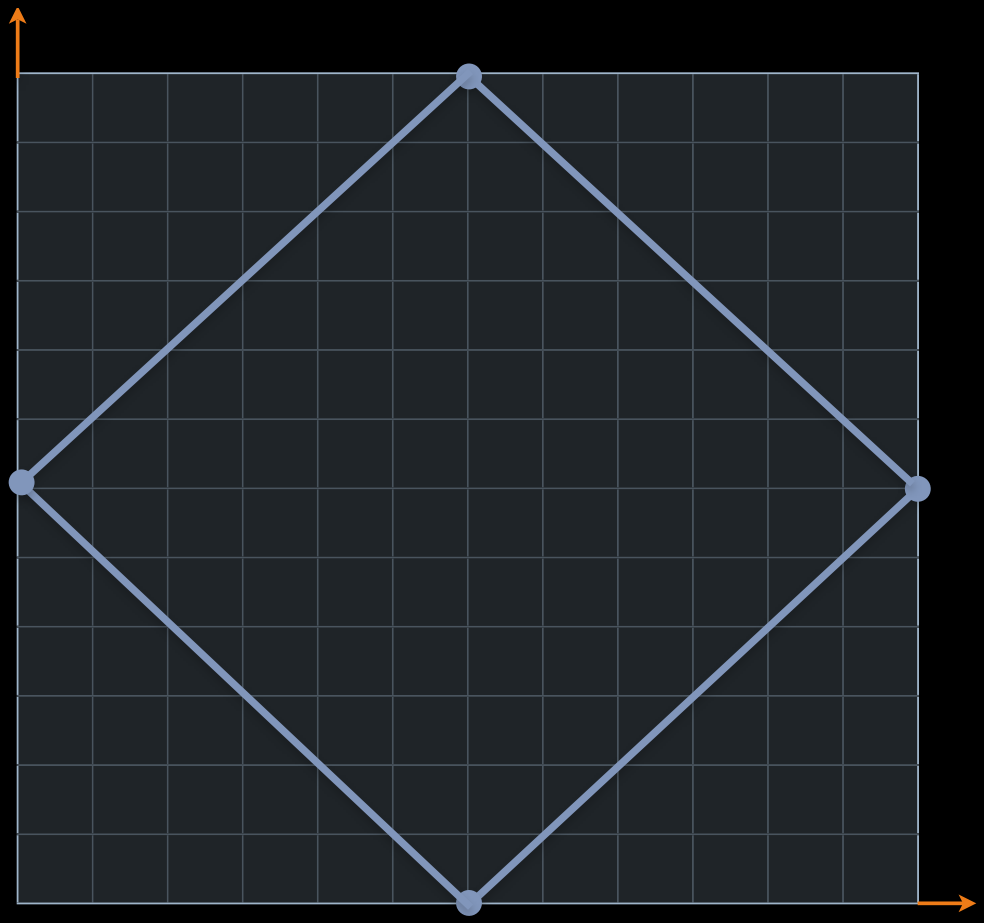


Basic Path Methods

```
- (void)moveToPoint:(NSPoint)point;
- (void)lineToPoint:(NSPoint)point;
- (void)closePath;
- (void)curveToPoint:(NSPoint)point
    controlPoint1:(NSPoint)c1
    controlPoint2:(NSPoint)c2;
```

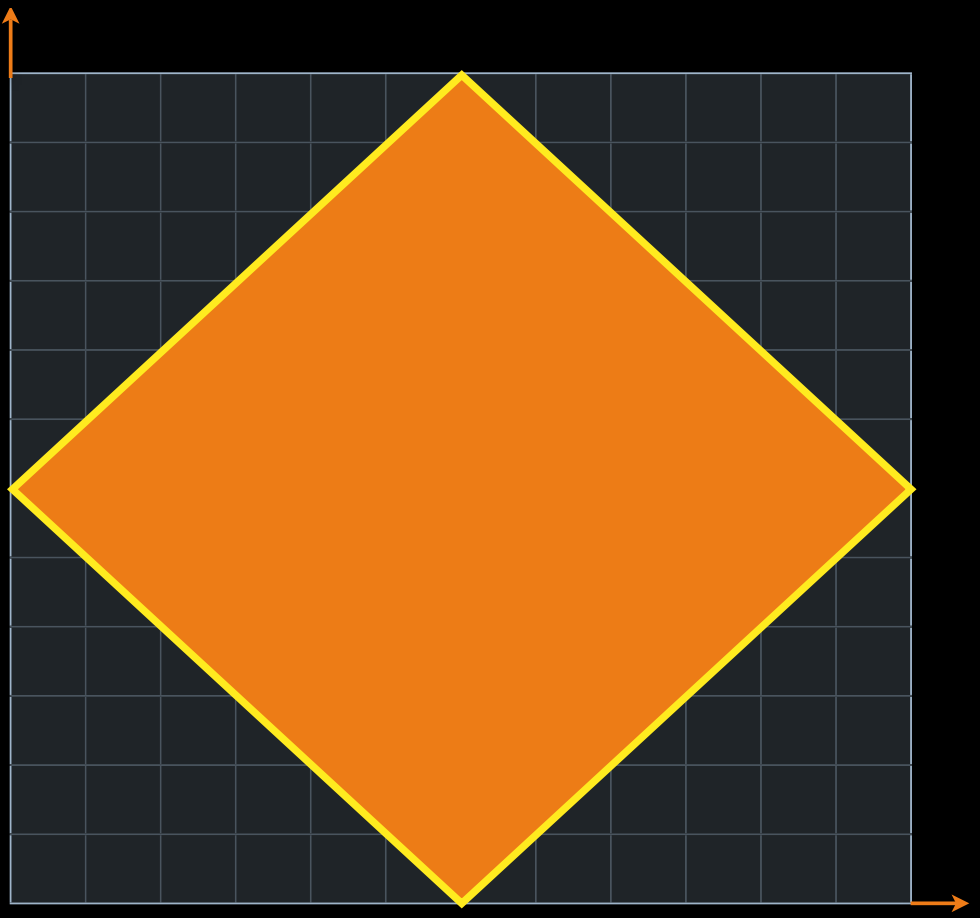
Build a Diamond

```
[path moveToPoint:
NSMakePoint(6, 0)];

[path addLineToPoint:
NSMakePoint(12, 6)];

[path addLineToPoint:
NSMakePoint(6, 12)];

[path addLineToPoint:
NSMakePoint(0, 6)];

[path closePath];
```



Draw the Diamond

```
[fillColor set];
[path fill];

[strokeColor set];
[path stroke];
```



Questions?