

Adding Apple IR Remote Support to Your App

Mark Aufflick

February 1, 2009



+



Rule #1 of Programming

- Everything should be code.
- Perhaps that's why I like \LaTeX so much.
 - (Or perhaps it's because MS Word and Endnote bite.)
- Ergo. . .

L^AT_EX slides–beamer

- The L^AT_EX beamer “document class” allows you to easily create slides with notes:

```
\begin{frame}{LaTeX slides--beamer}
  \begin{itemize}
    \item The LaTeX beamer “document class”
      allows you to easily create slides with notes:
  \end{itemize}
\end{frame}

\note[itemize]{
  \item Here are some notes.
}
```

SplitShow PDF Presentation

- `pdflatex` + `beamer` → PDF files
- So we need a way to present PDF files
- Preferably with notes on our second screen so I can sound like I know what I'm talking about
- Splitshow.app allows two screen presentations from PDF
- <http://code.google.com/p/splitshow>

2009-02-01

Adding Apple IR Remote Support to Your App

└ SplitShow PDF Presentation

SplitShow PDF Presentation

- `pdflatex + beamer` → PDF files
- So we need a way to present PDF files
- Preferably with notes on our second screen so I can sound like I know what I'm talking about
- Splitshow.app allows two screen presentations from PDF
- <http://code.google.com/p/splitshow>

- Toggle to notes screen by holding Play/Pause button on remote.

Apple IR Remote Support

- Splitshow didn't have Apple IR Remote Support
- I added it using classes by Martin Kahr:
- <http://www.martinkahr.com/source-code>

Implementing Apple IR Remote Support

- Add the class files to your project
- Add the Carbon Framework (the IR classes dig into IOKit and Carbon)
- Create a class to be a delegate to both `NSApplication` and `RemoteControl`
 - Or you can use `RemoteControlContainer` instead of `RemoteControl` and get `Keyspan` support for free

NSApplication Delegate Methods

Create `RemoteControl` instance, and set self as delegate via:

```
- (void)applicationDidFinishLaunching:(NSNotification *)n
{
    remoteControl = [[RemoteControl alloc] initWithDelegate: self];
    [remoteControl startListening: self];
}
```

Monopolise/release the remote control via:

```
- (void)applicationWillBecomeActive:(NSNotification *)n
{
    [remoteControl startListening: self];
}
- (void)applicationWillResignActive:(NSNotification *)n
{
    [remoteControl stopListening: self];
}
```

Remember to instantiate and connect your `NSApplication` delegate in `MainMenu.xib`

NSApplication Delegate Methods

Create `RemoteControl` instance, and set self as delegate via:

```
- (void)applicationDidFinishLaunching:(NSNotification *)n
{
    remoteControl = [[RemoteControl alloc] initWithDelegate: self];
    [remoteControl startListening: self];
}
```

Monopolise/release the remote control via:

```
- (void)applicationWillBecomeActive:(NSNotification *)n
{
    [remoteControl startListening: self];
}
- (void)applicationWillResignActive:(NSNotification *)n
{
    [remoteControl stopListening: self];
}
```

Remember to instantiate and connect your `NSApplication` delegate in `MainMenu.xib`

UIApplication Delegate Methods

Create RemoteControl instance, and set self as delegate via:

```
- (void)applicationDidFinishLaunching:(NSNotification *)n
{
    remoteControl = [[RemoteControl alloc] initWithDelegate: self];
    [remoteControl startListening: self];
}
```

Monopolise/release the remote control via:

```
- (void)applicationWillBecomeActive:(NSNotification *)n
{
    [remoteControl startListening: self];
}
- (void)applicationWillResignActive:(NSNotification *)n
{
    [remoteControl stopListening: self];
}
```

Remember to instantiate and connect your UIApplication delegate in MainMenu.xib

- <http://code.google.com/p/splitshow/source/browse/trunk/src/SSAppDelegate.m>
- Switch to Xcode to show code (after revealing second method).

RemoteControl Delegate Method

```
- (void) sendRemoteButtonEvent:(RemoteControlEventsIdentifier) event  
    pressedDown:(BOOL) pressedDown  
    remoteControl:(RemoteControl *) remoteControl
```

- event is an integer, matching one of a number of constants, eg:
 - kRemoteButtonLeft
 - kRemoteButtonRight
 - kRemoteButtonRight_Hold
 - kRemoteButtonPlay
- Simply perform the appropriate method call by matching event in a case statement.
- Often you want to discard button up events.

RemoteControl Delegate Method

RemoteControl Delegate Method

```
- (void)sendRemoteButtonEvent:(RemoteControlEventsIdentifier)event  
    pressedDown:(BOOL)pressedDown  
    remoteControl:(RemoteControl *)remoteControl
```

- event is an integer, matching one of a number of constants, eg:
 - kRemoteButtonLeft
 - kRemoteButtonRight
 - kRemoteButtonRight_Hold
 - kRemoteButtonPlay
- Simply perform the appropriate method call by matching event in a case statement.
- Often you want to discard button up events.

- <http://code.google.com/p/splitshow/source/browse/trunk/src/SSAppDelegate.m>
- Switch to Xcode to show code.

Fin

- Now your Cocoa Application has Apple Remote support :)