CS193P - Lecture 11

iPhone Application Development

Text Input
Presenting Content Modally

Announcements

- Presence 3 assignment has been posted, due Tuesday 5/12
- **Final project proposals** due on Monday 5/11

Announcements

- This week's bonus section with Steve Marmon
  - Discussing interface design for iPhone apps
  - Will be available on iTunes U

Today’s Topics

- Using the Clang Static Analyzer to find bugs
- iPhone Keyboards
- Customizing Text Input
- Presenting Content Modally

Finding Bugs with Clang Static Analyzer

- Tool for static analysis of C/Objective-C code
- Identifies potential bugs
  - Leaks
  - Using uninitalized or released variables
  - Missing dealloc method
  - More...
- Early in development, watch out for false positives
- 100% open source!
- More info at http://clang.llvm.org/StaticAnalysis.html

Running the Clang Static Analyzer

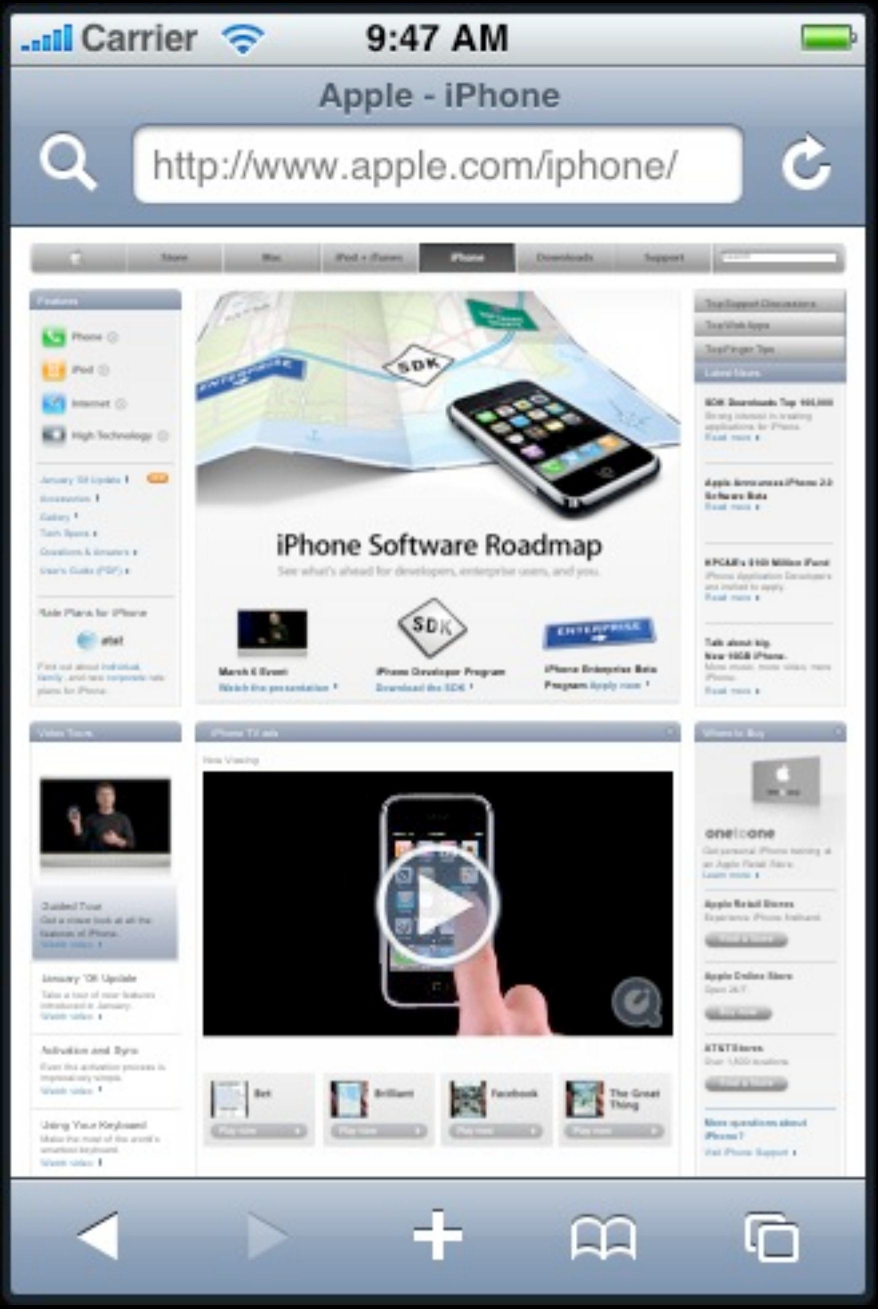
- Clean build in Xcode first
  - Ensure that nothing gets left out
- From the command line in your project directory:
  - `scan-build -k -V xcodebuild -configuration Debug -sdk iphonesimulator2.2`
  - (Customize as needed)
- Results open up in Safari when completed!

Demo:
Using the Clang Static Analyzer

iPhone Keyboards

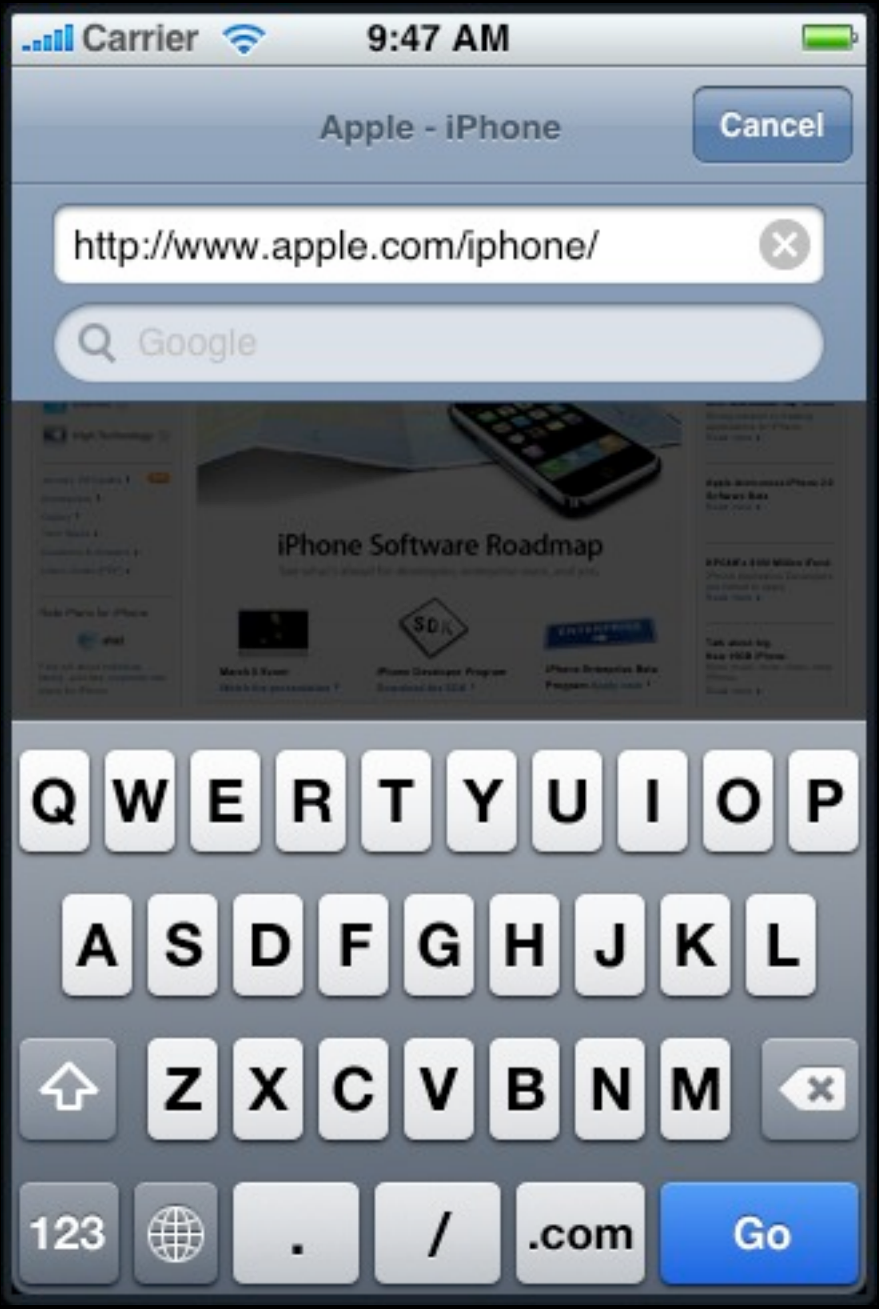
Virtual keyboard

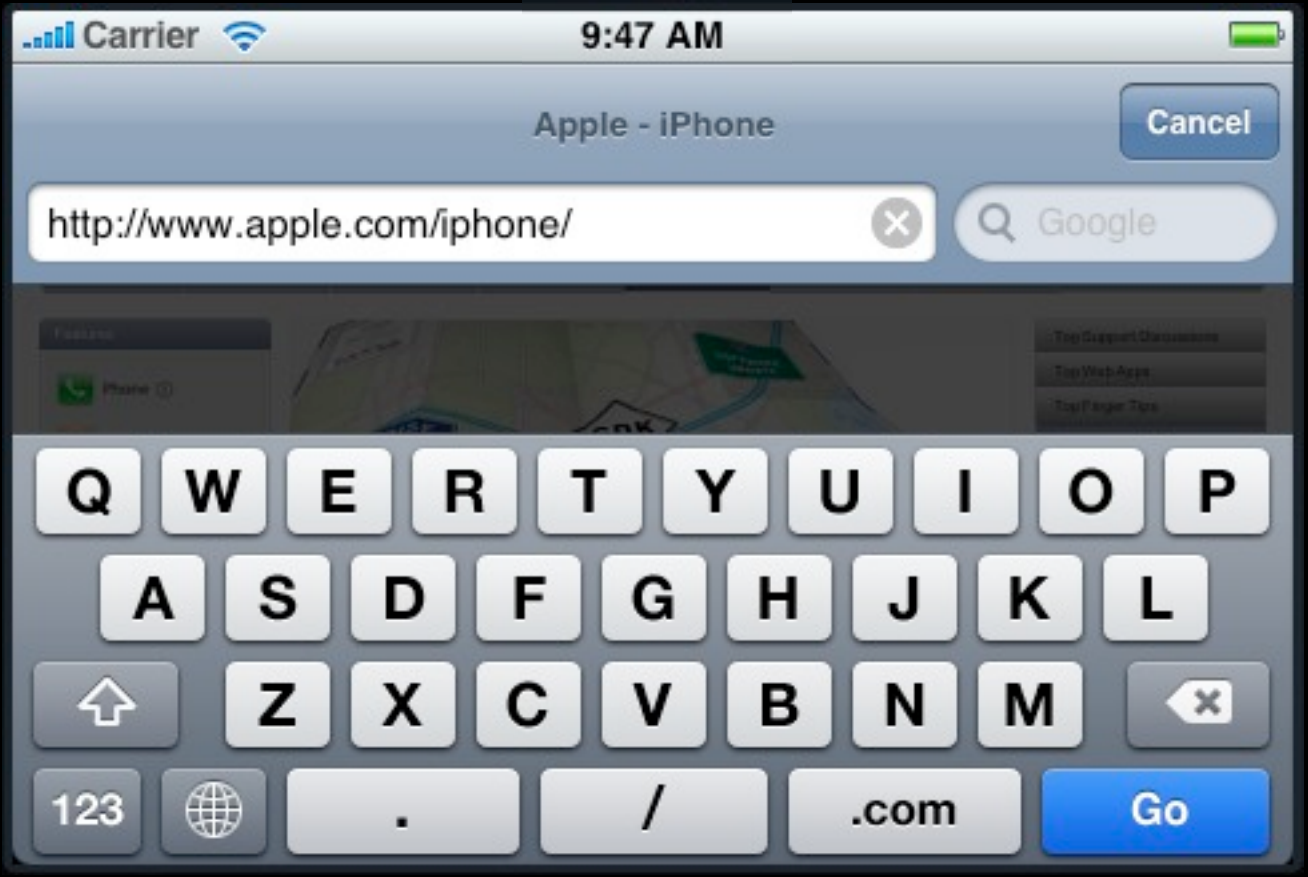
Appears when needed



Virtual keyboard

Appears when needed

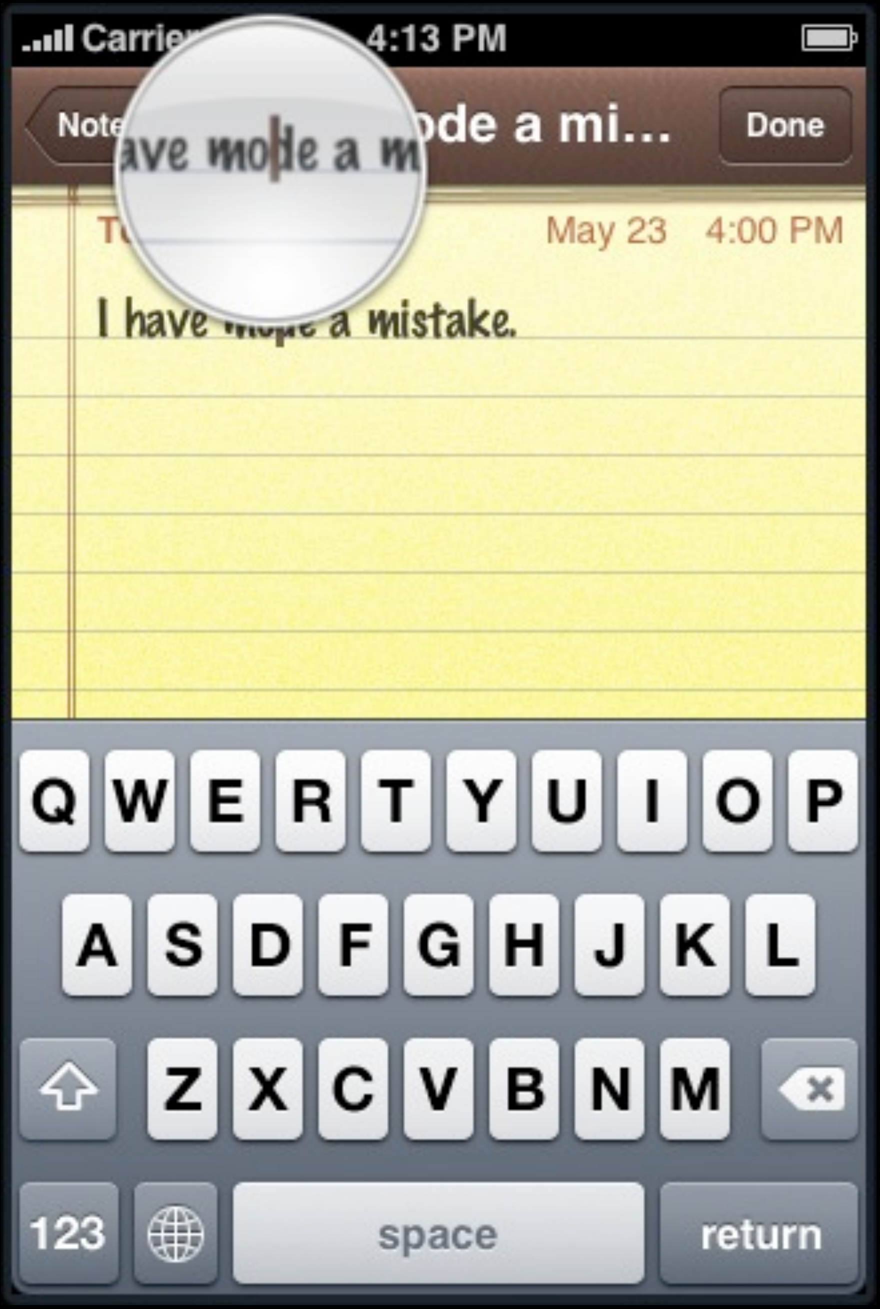




Portrait and Landscape

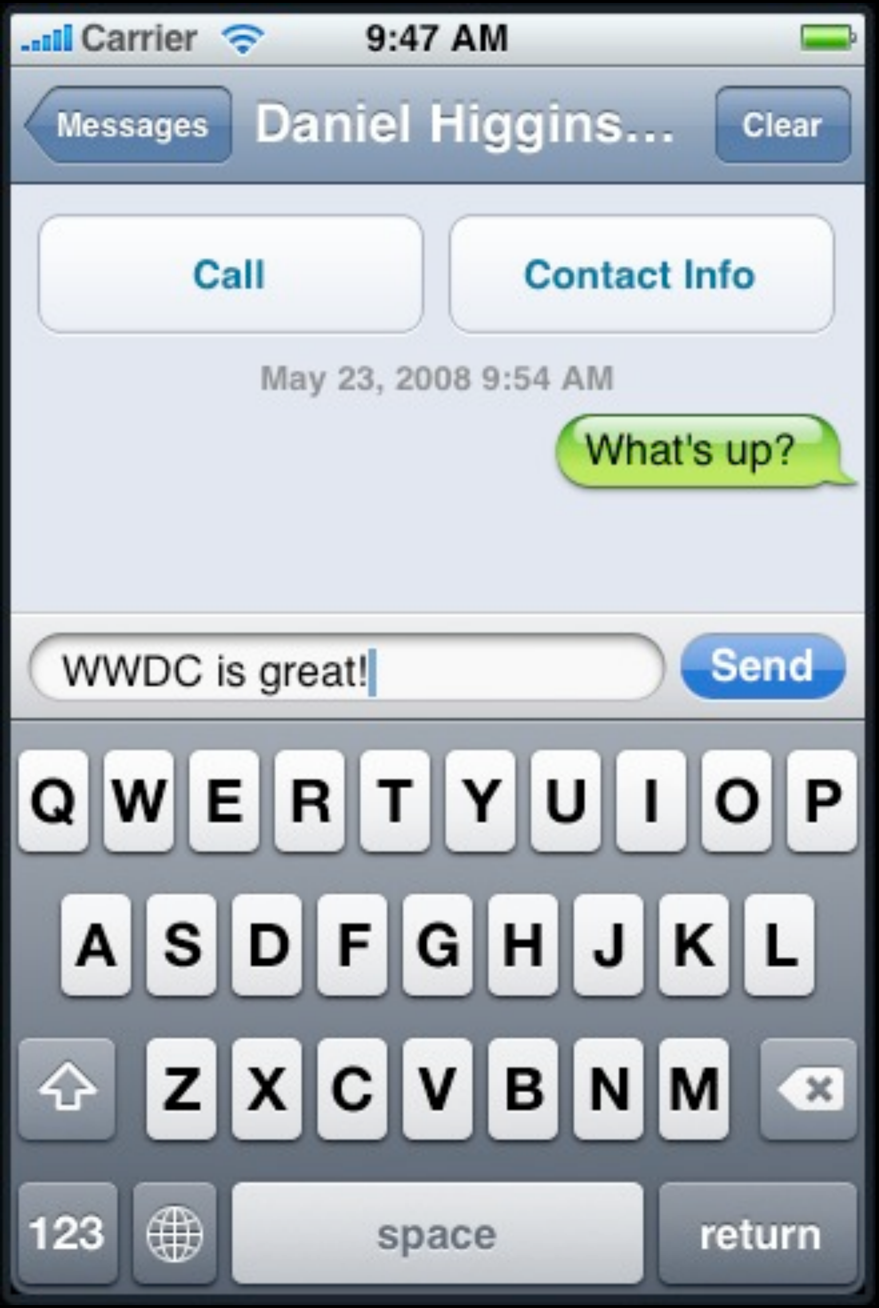
Simple selection model

Text loupe/magnifier



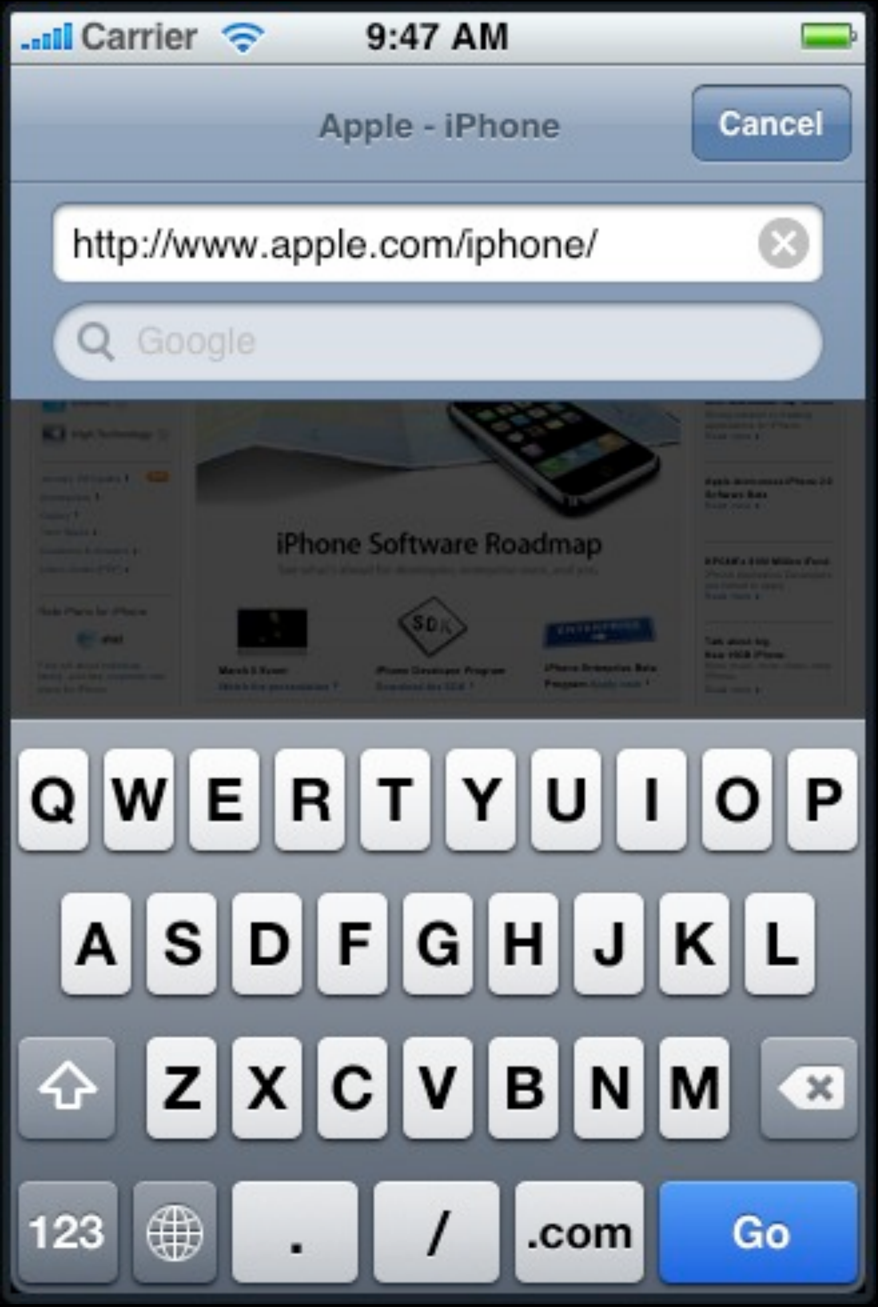
Many keyboard types

Adapted to task



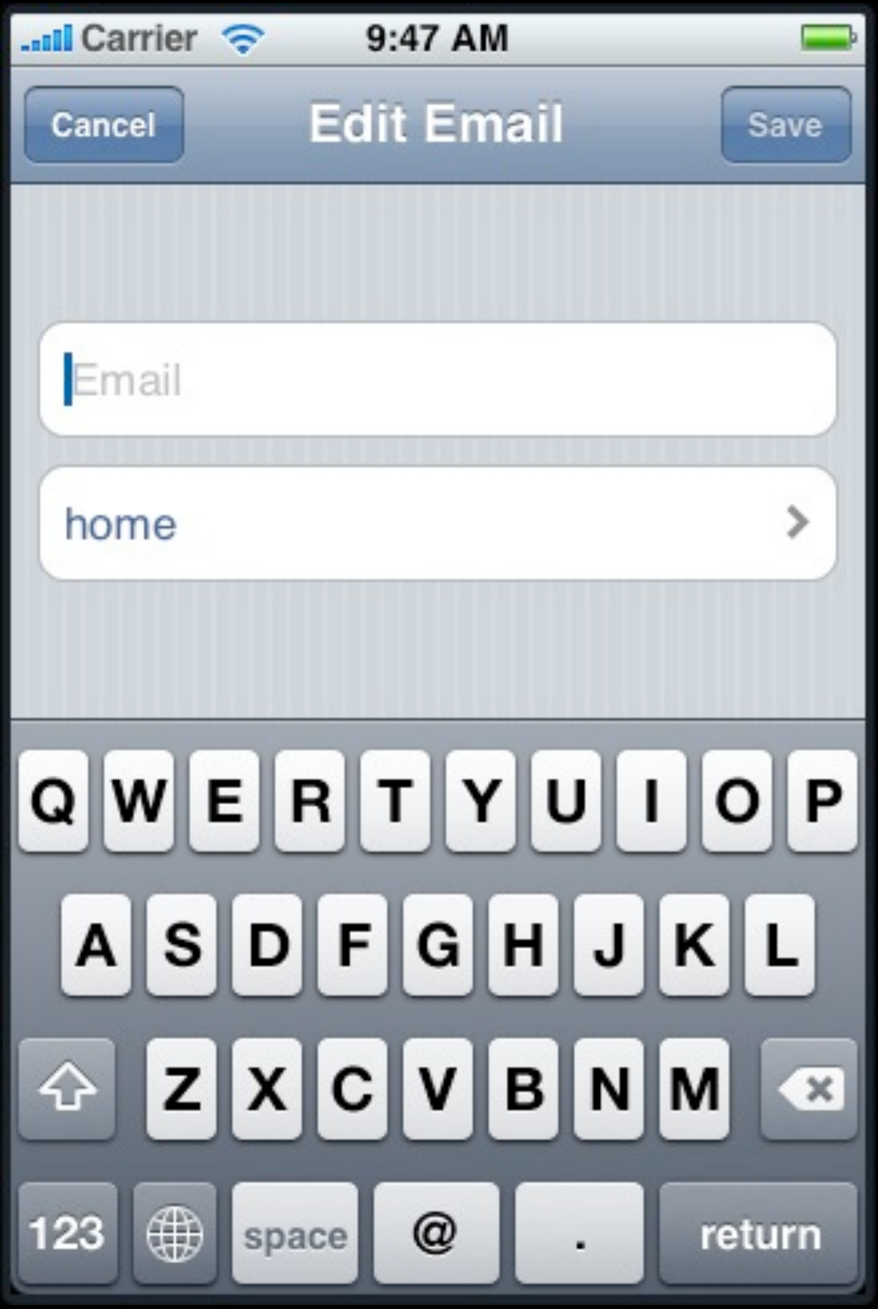
Many keyboard types

Adapted to task



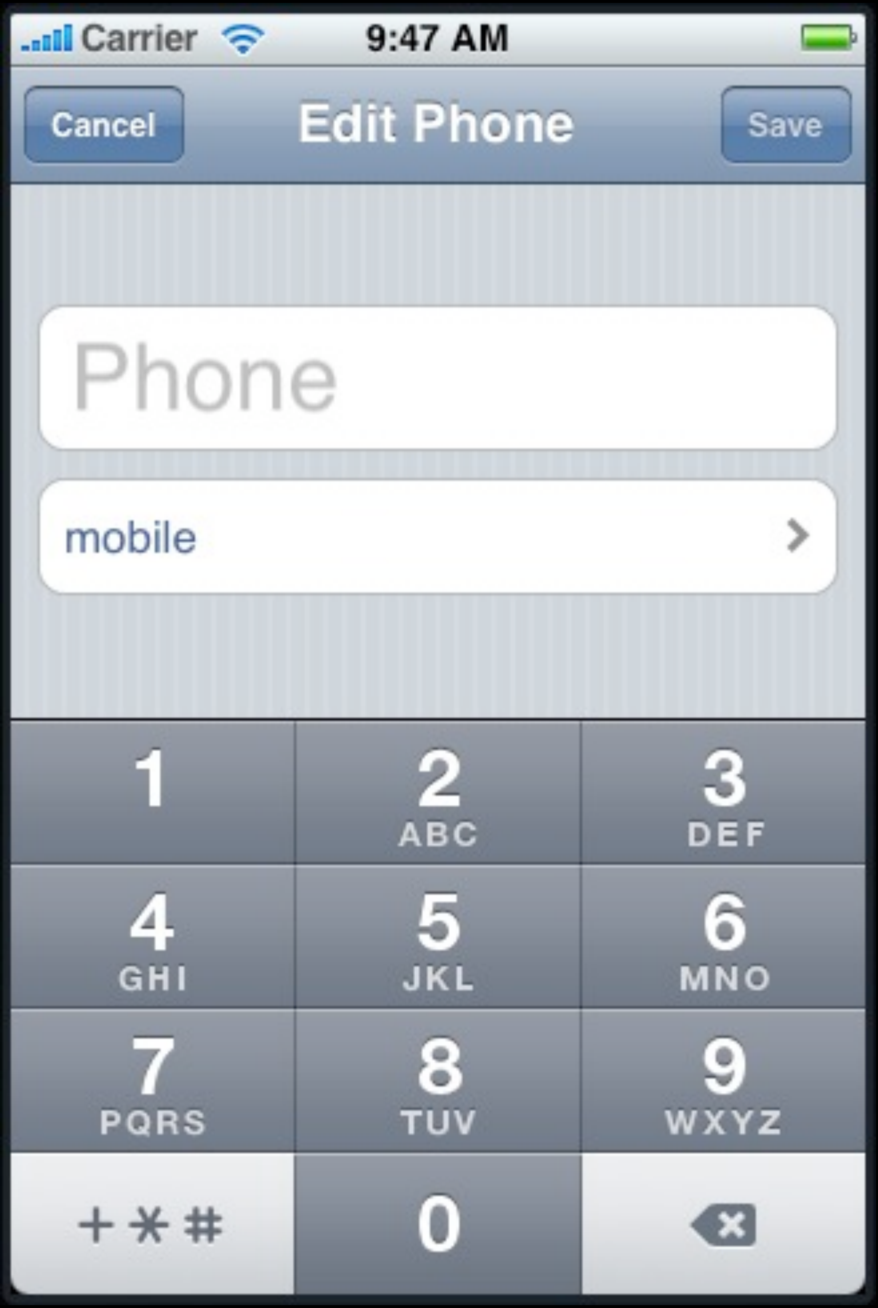
Many keyboard types

Adapted to task



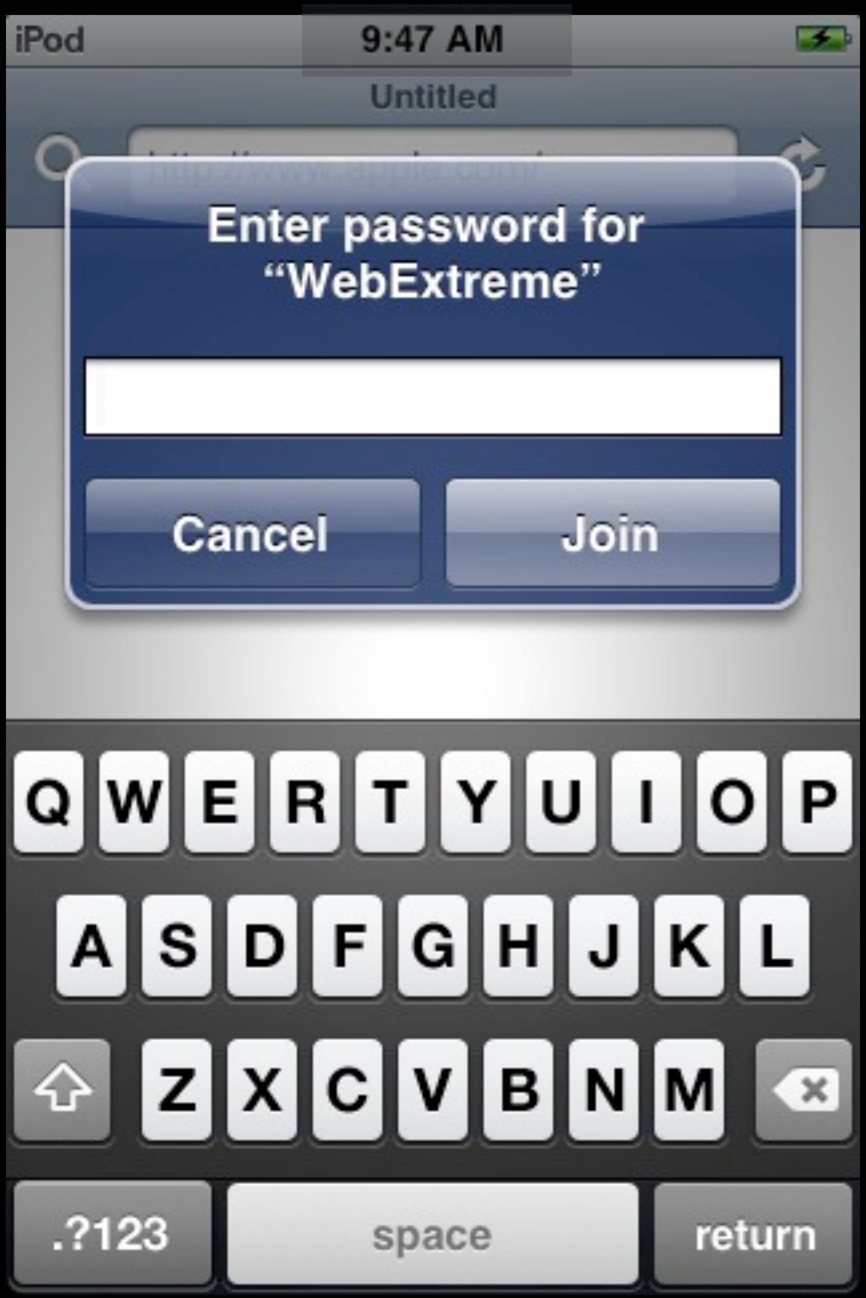
Many keyboard types

Adapted to task

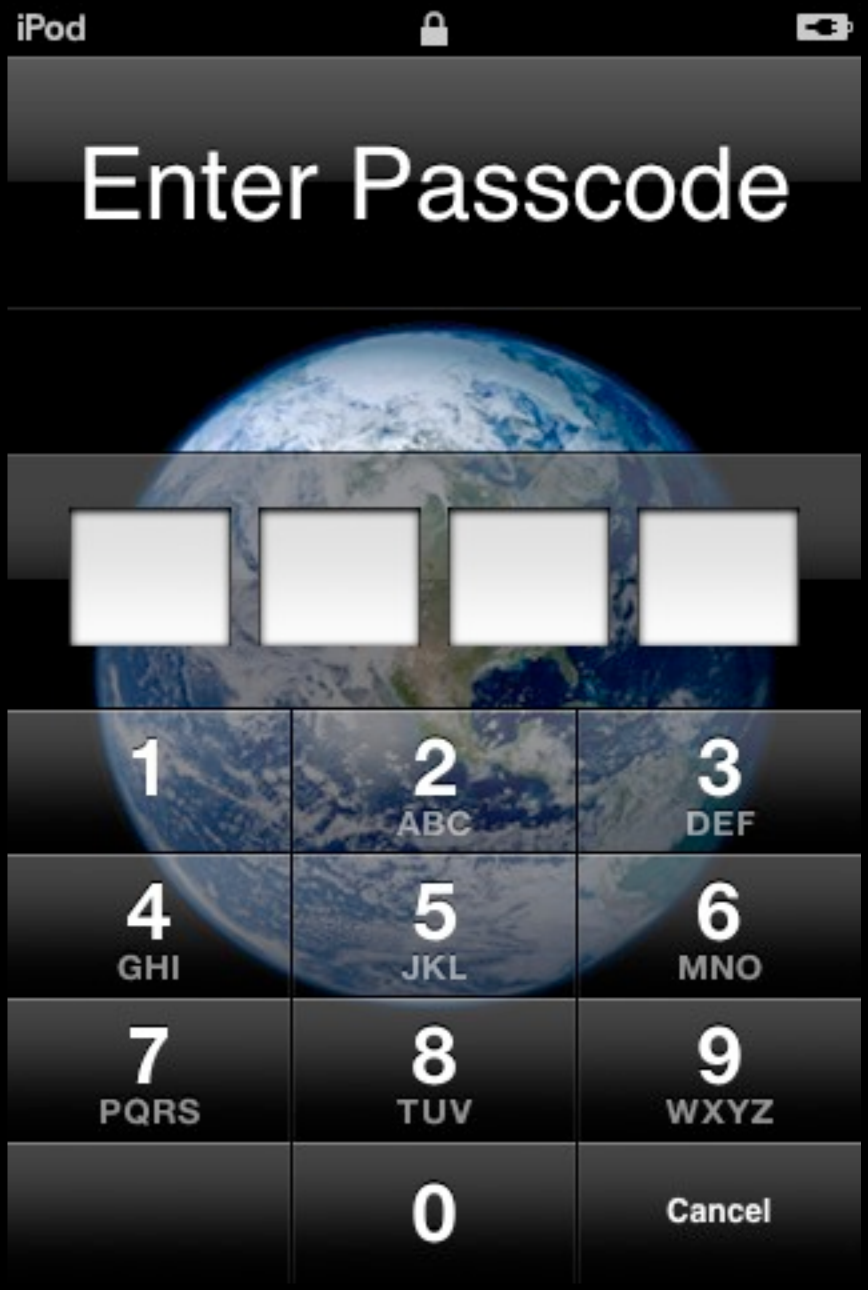


Many keyboard types

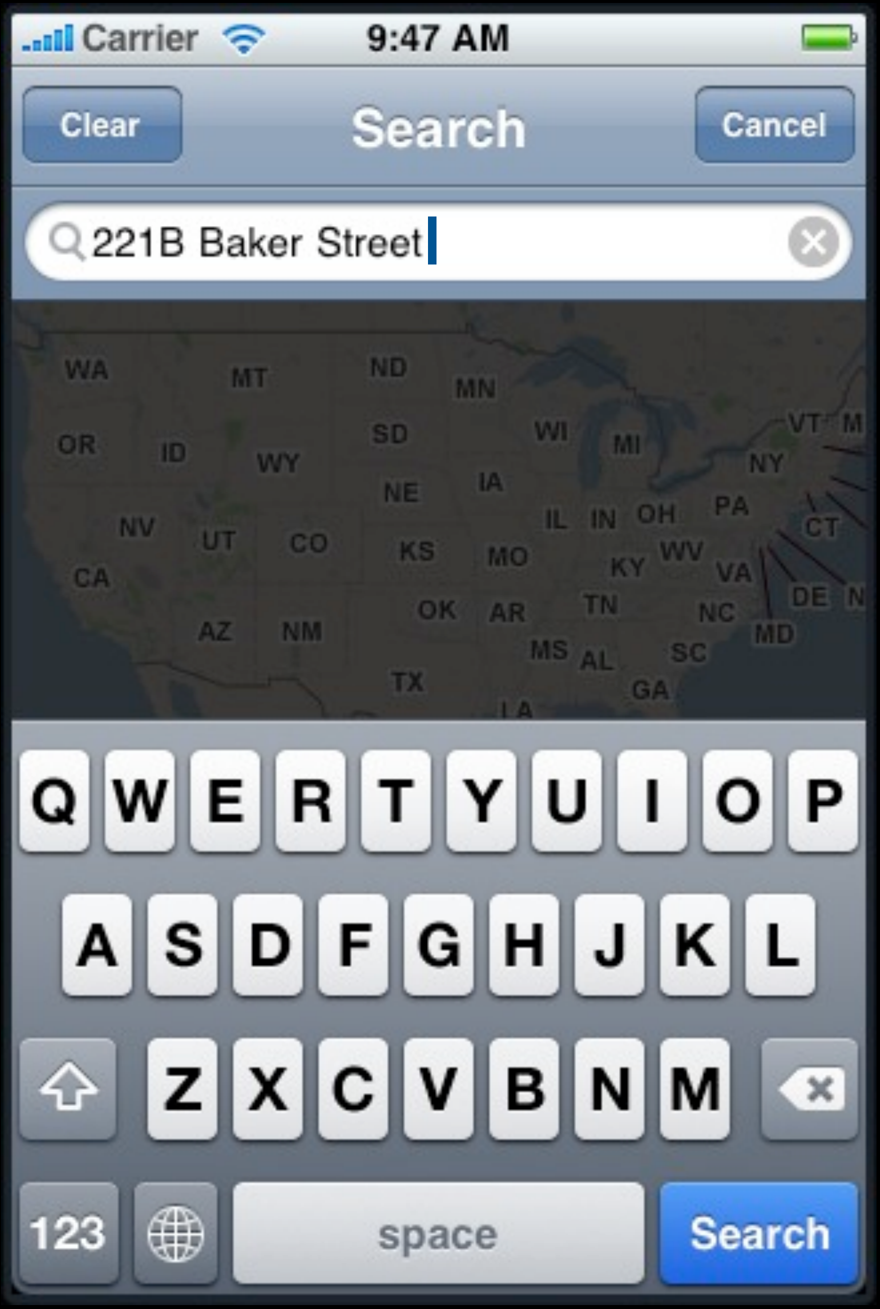
Adapted to task



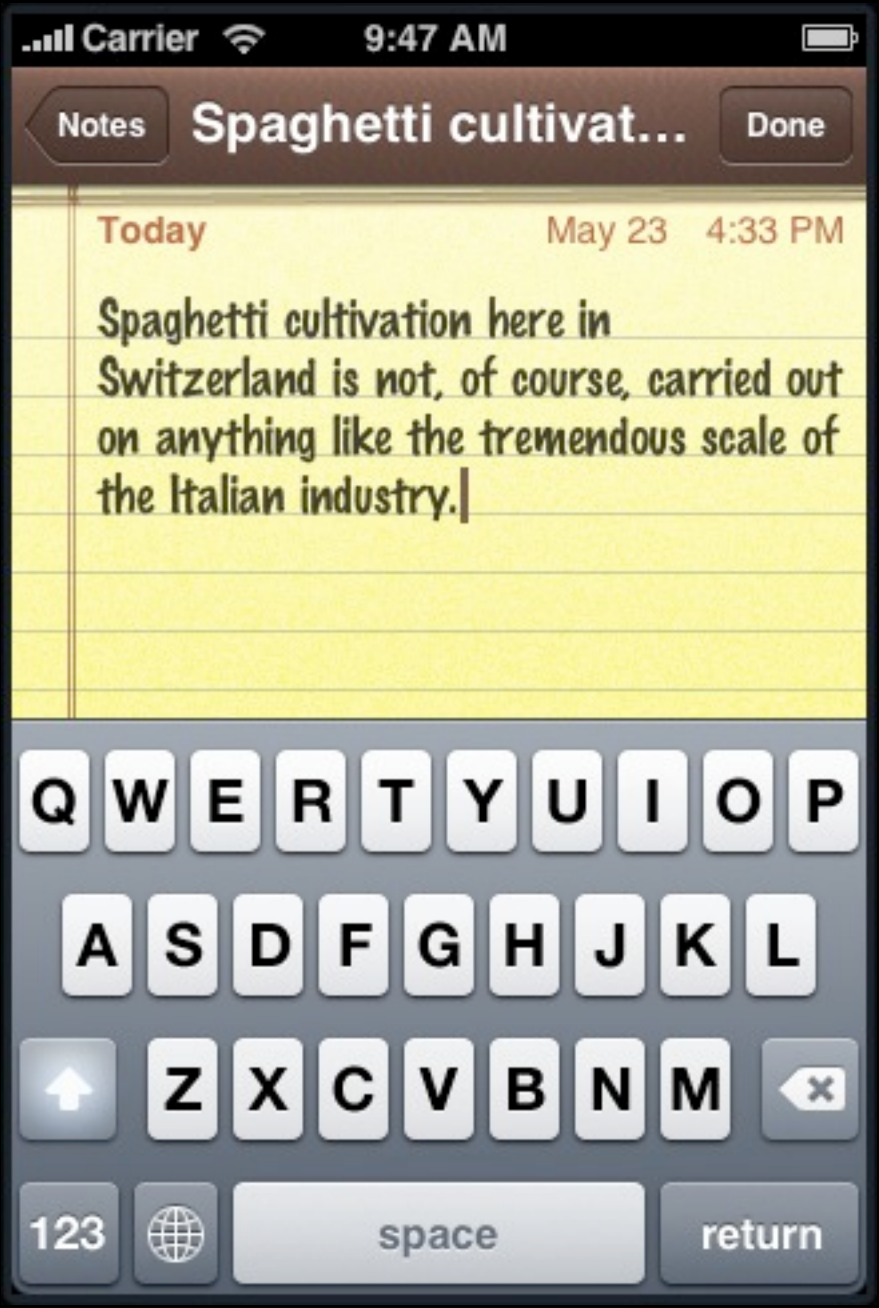
Many keyboard types
Adapted to task



Single line editing



Multi-line editing

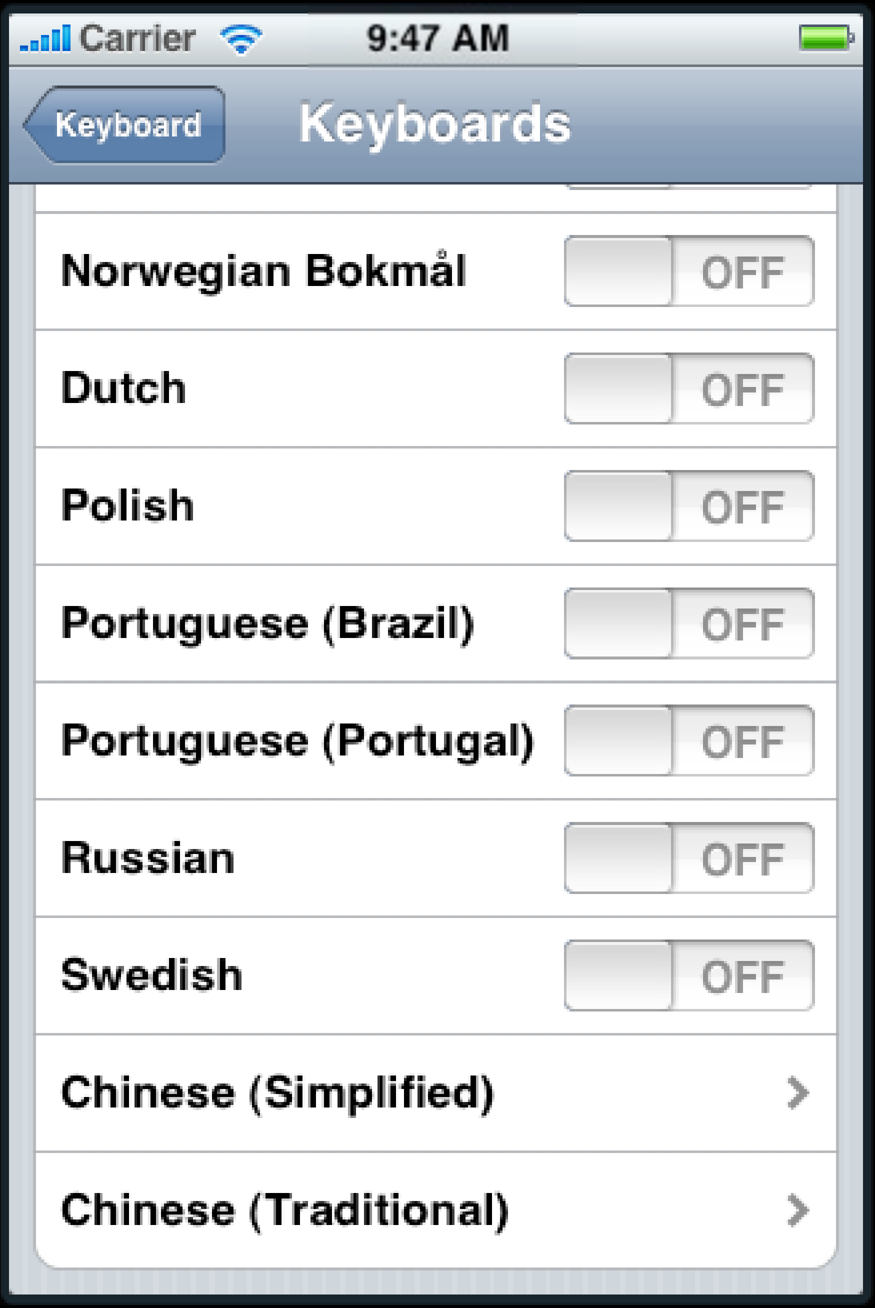


Spaghetti cultivation here in Switzerland is not, of course, carried out on anything like the tremendous scale of the Italian industry.

20

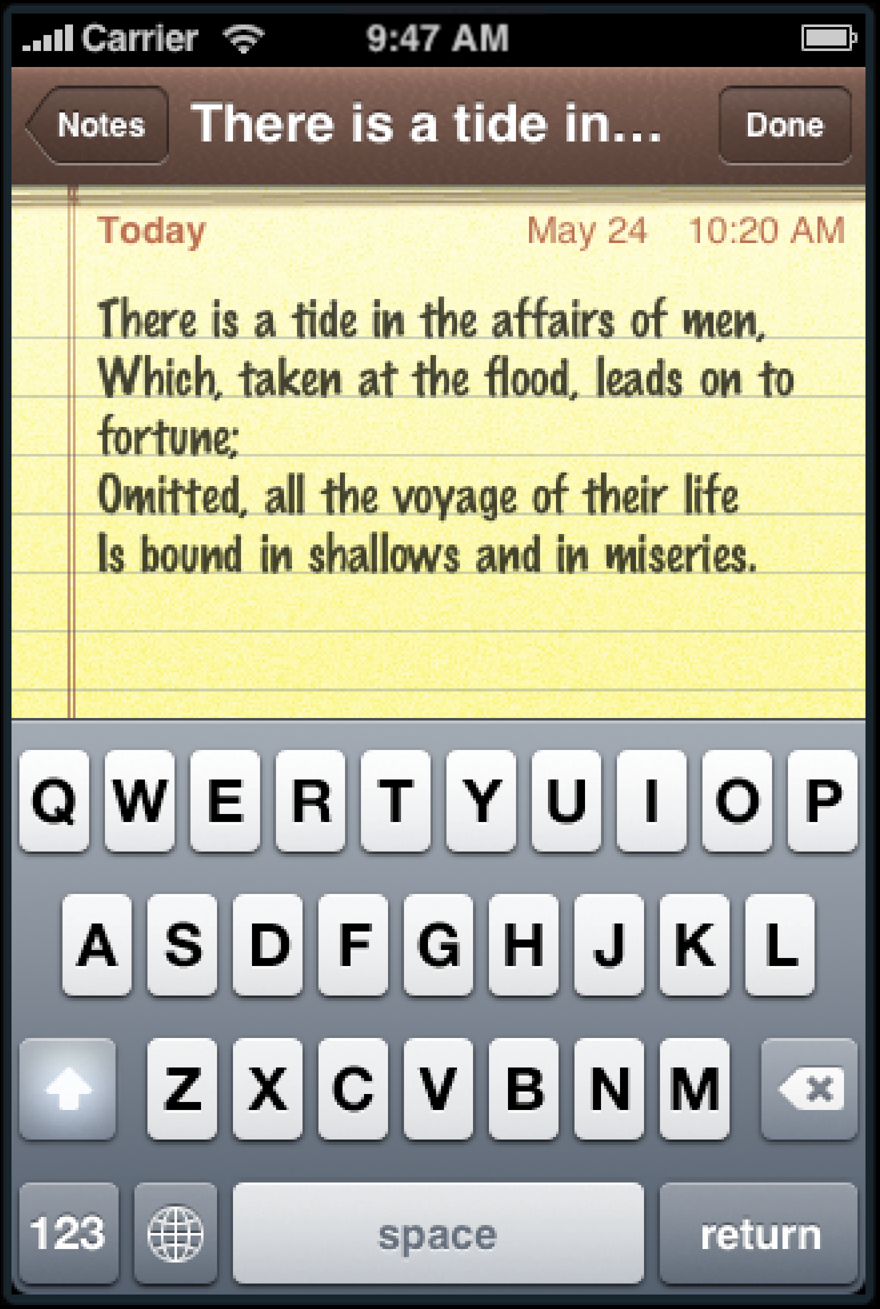
Languages

Full dictionary support



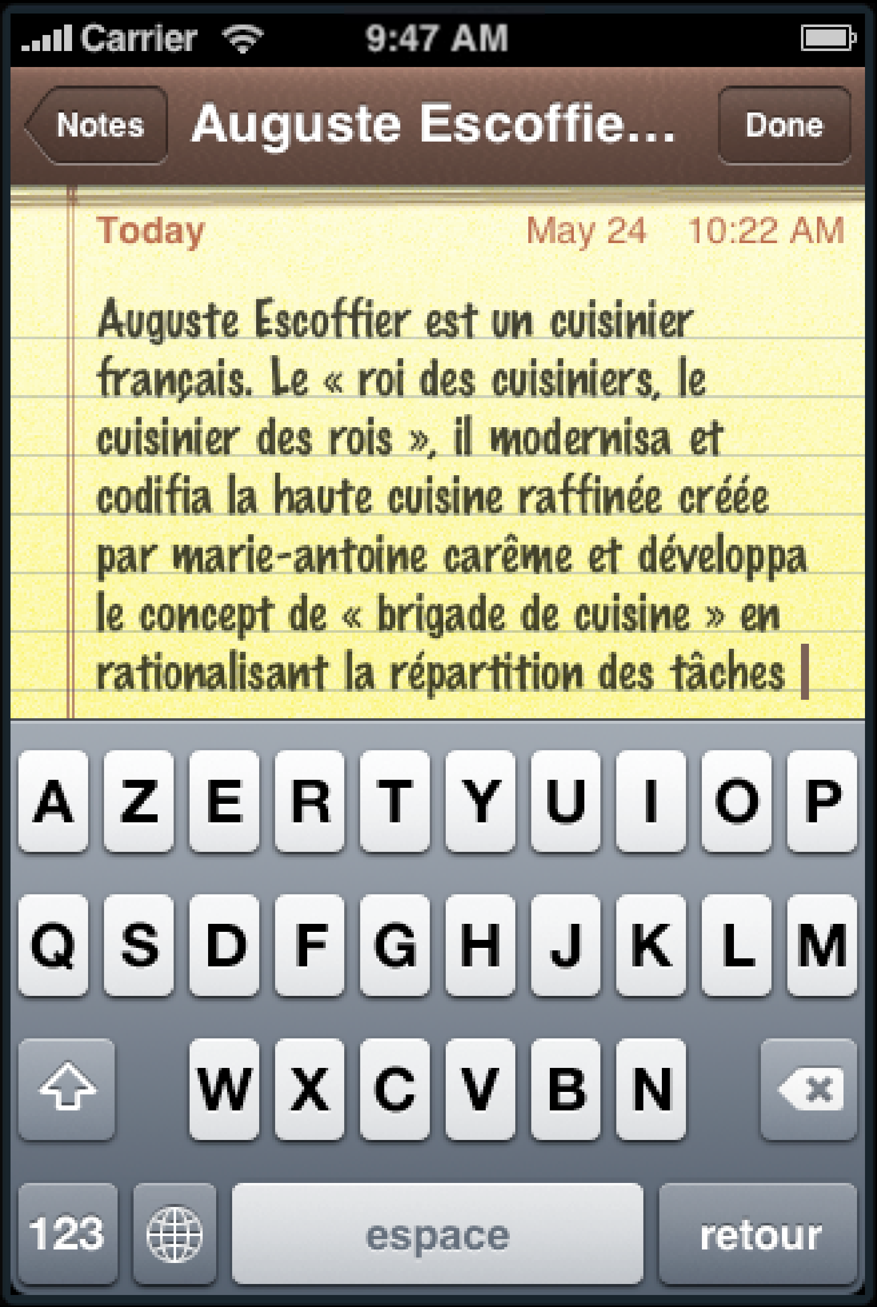
Norwegian Bokmål	OFF
Dutch	OFF
Polish	OFF
Portuguese (Brazil)	OFF
Portuguese (Portugal)	OFF
Russian	OFF
Swedish	OFF
Chinese (Simplified)	>
Chinese (Traditional)	>

English



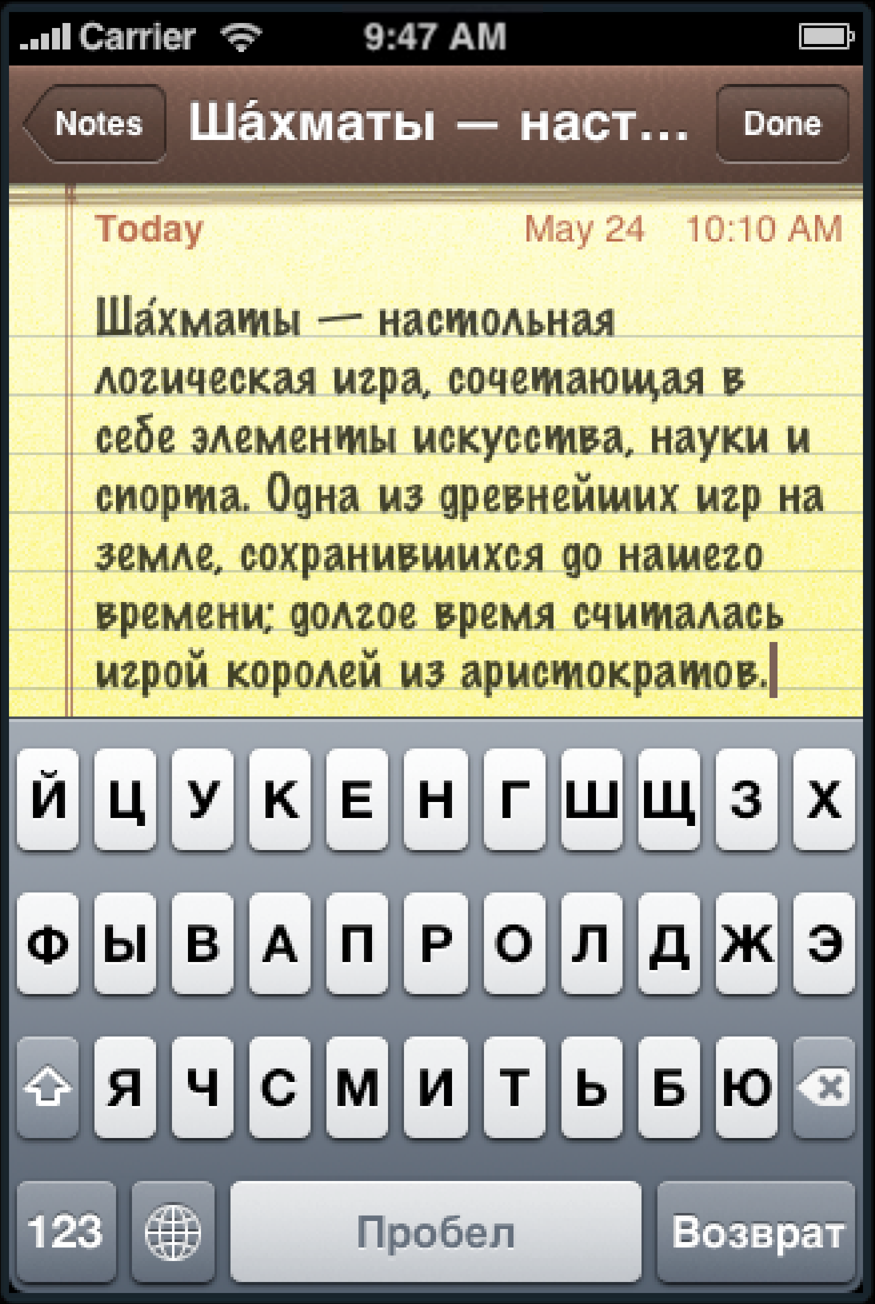
There is a tide in the affairs of men,
Which, taken at the flood, leads on to fortune;
Omitted, all the voyage of their life
Is bound in shallows and in miseries.

French



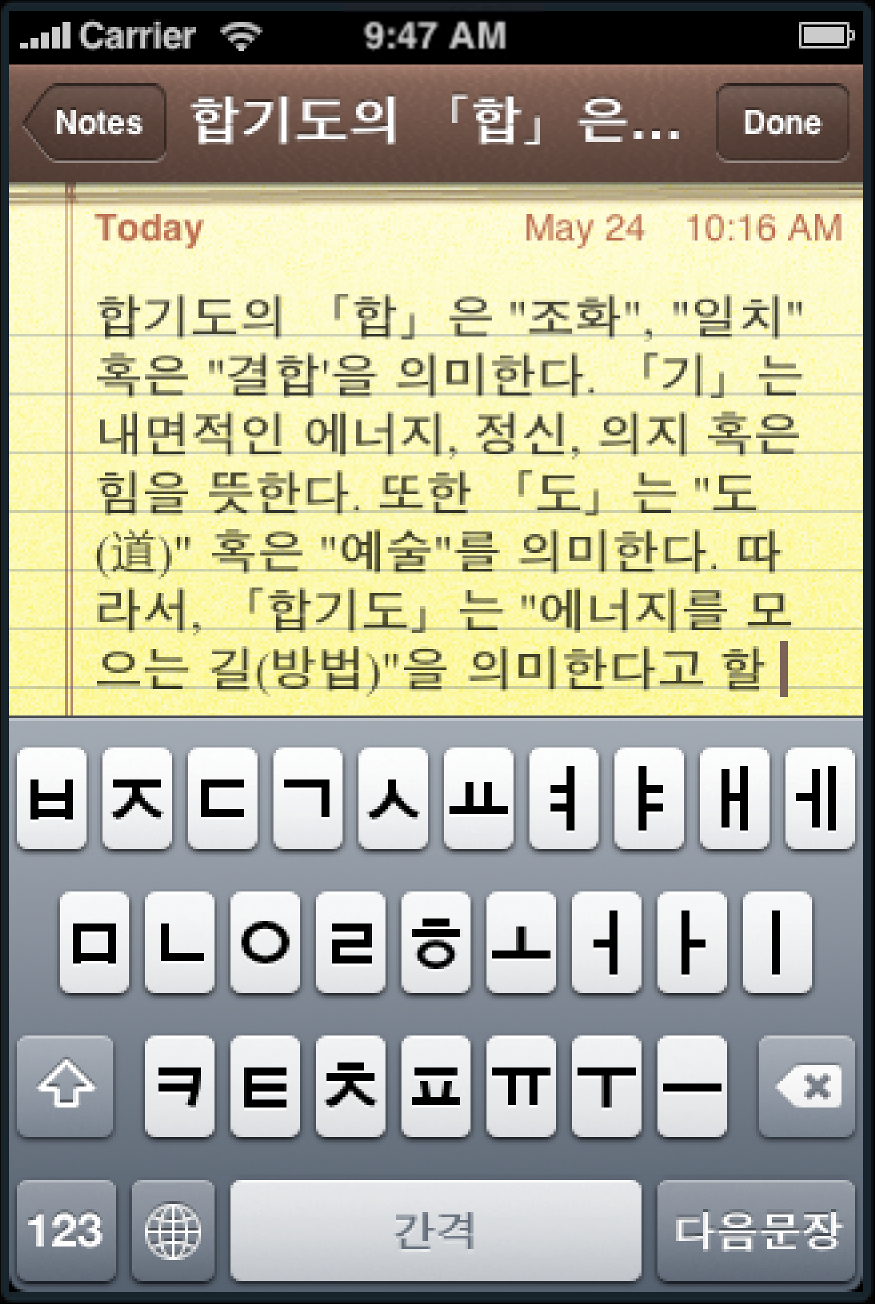
Auguste Escoffier est un cuisinier français. Le « roi des cuisiniers, le cuisinier des rois », il modernisa et codifia la haute cuisine raffinée créée par marie-antoine carême et développa le concept de « brigade de cuisine » en rationalisant la répartition des tâches

Russian



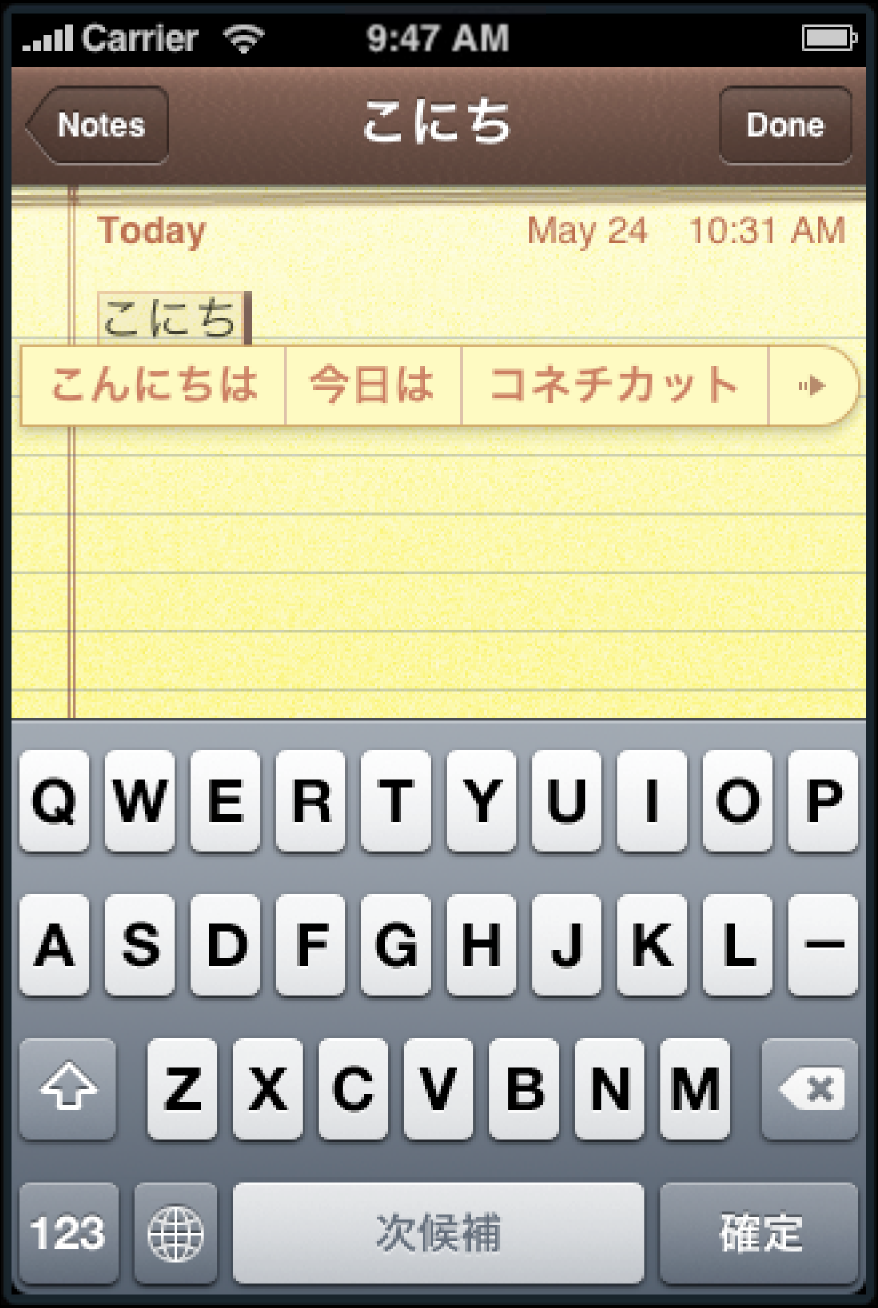
Шáхматы — настольная логическая игра, сочетающая в себе элементы искусства, науки и спорта. Одна из древнейших игр на земле, сохранившихся до нашего времени; долгое время считалась игрой королей из аристократов.

Korean

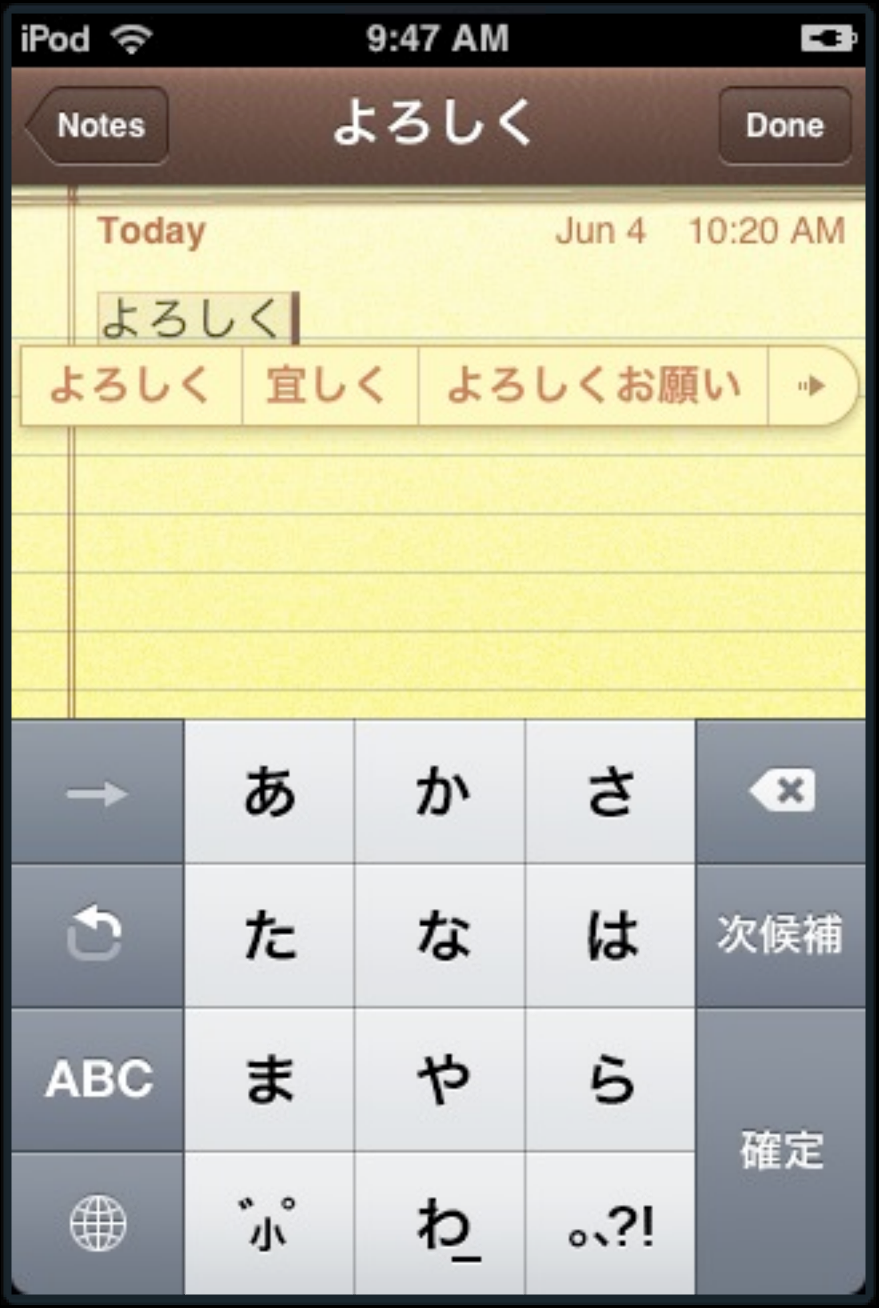


합기도의 「합」 은 "조화", "일치" 혹은 "결합'을 의미한다. 「기」 는 내면적인 에너지, 정신, 의지 혹은 힘을 뜻한다. 또한 「도」 는 "도(道)" 혹은 "예술"를 의미한다. 따라서, 「합기도」 는 "에너지를 모으는 길(방법)"을 의미한다고 할

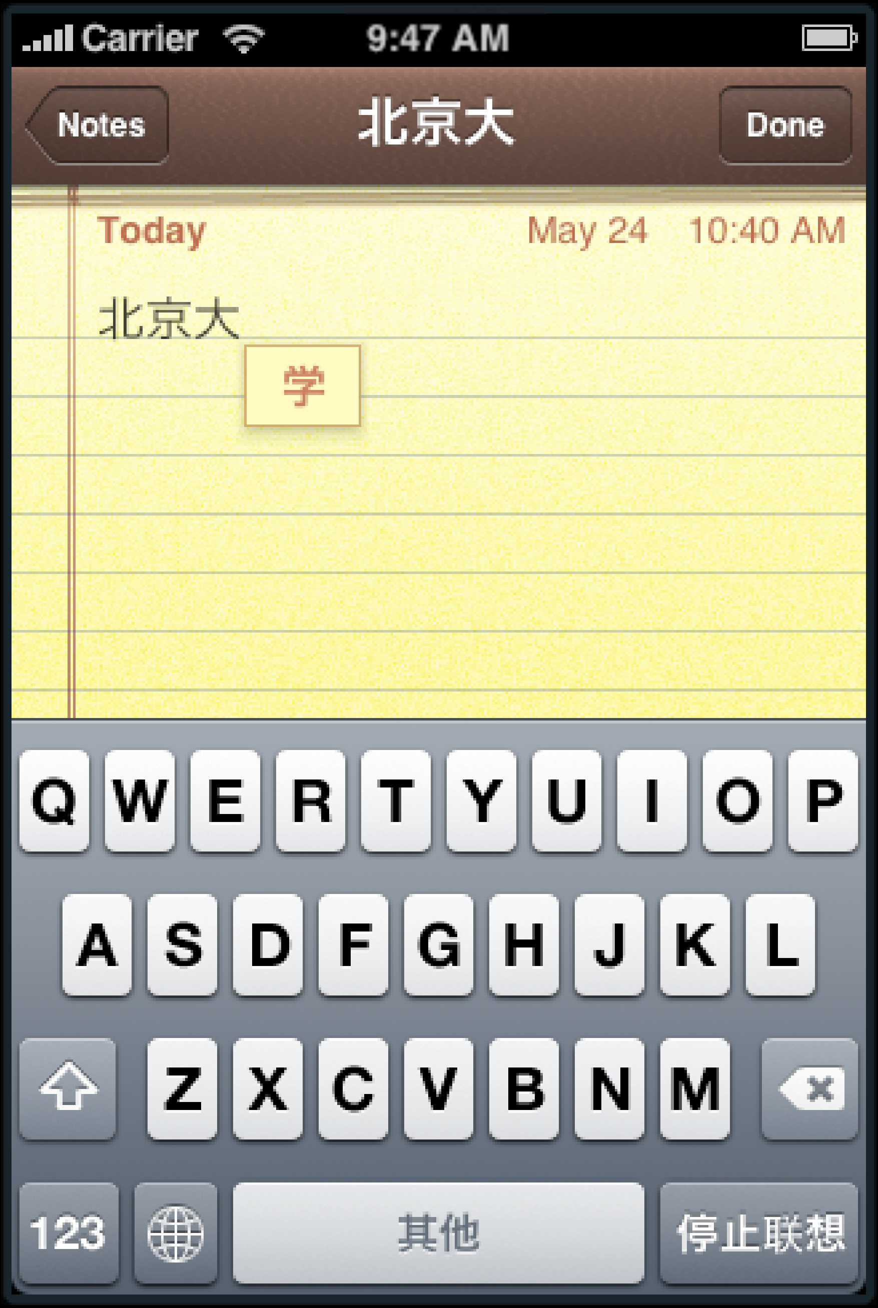
Japanese Romaji



Japanese Kana



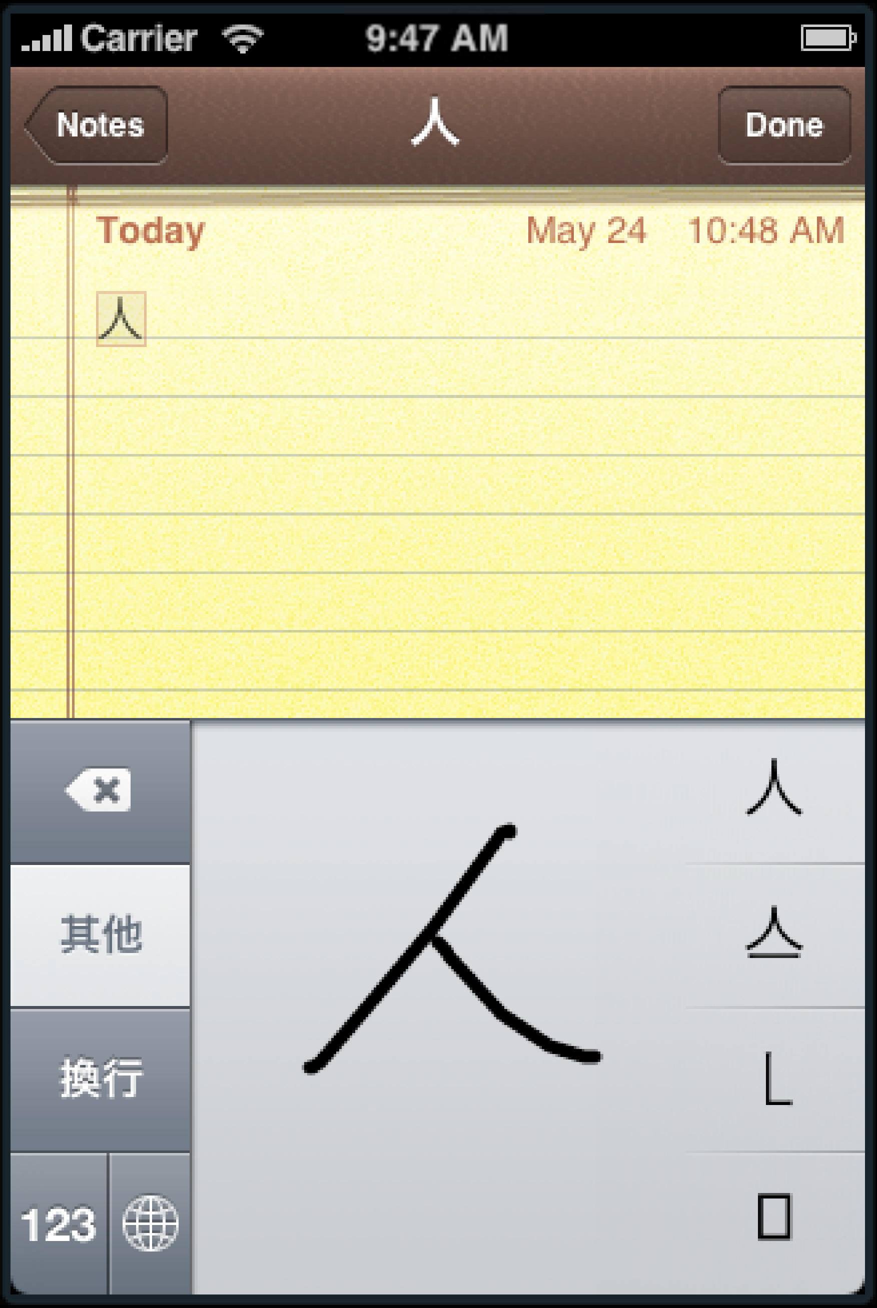
Chinese Pinyin



Chinese Handwriting

Simplified

Traditional



Customizing Text Input

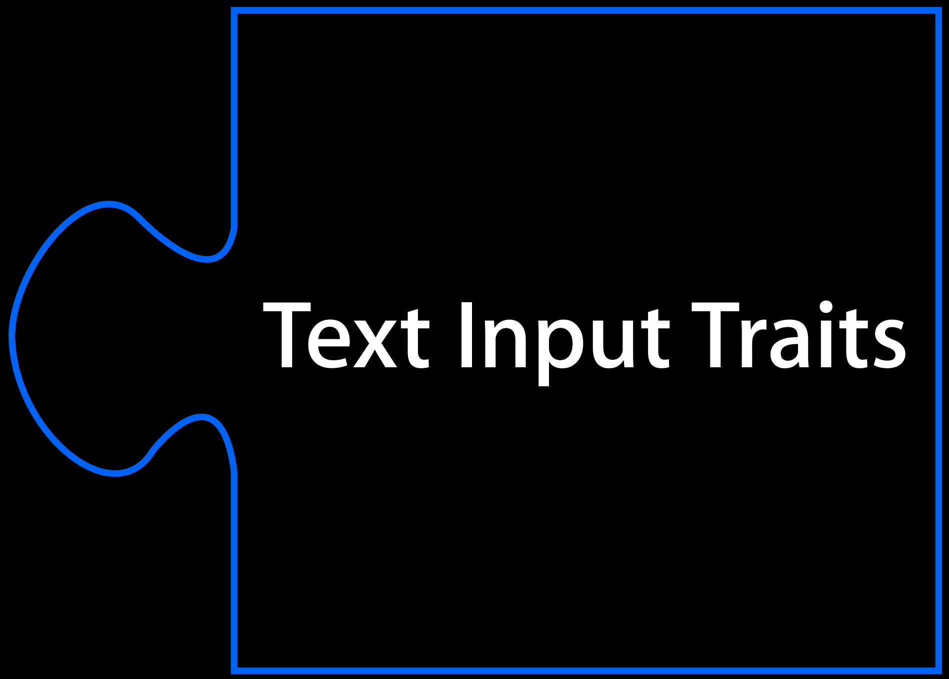
Text Containers

Text Containers

Delegates
Notifications
Methods

Text Containers

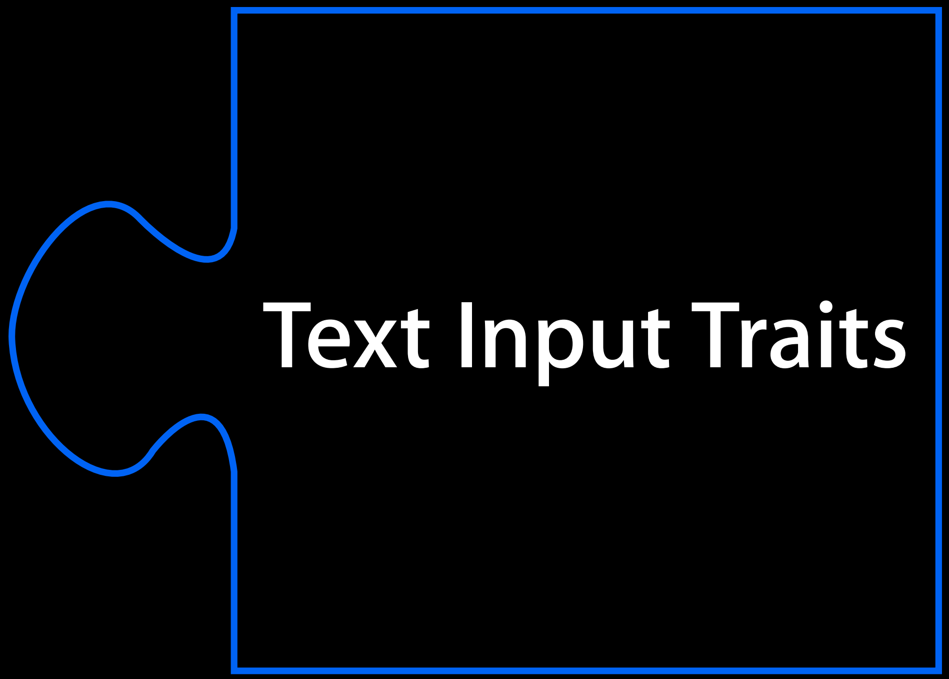
Text Input Traits



Protocol

UITextField

UITextView



Autocapitalization

Autocorrection

Keyboard Type

Keyboard Appearance

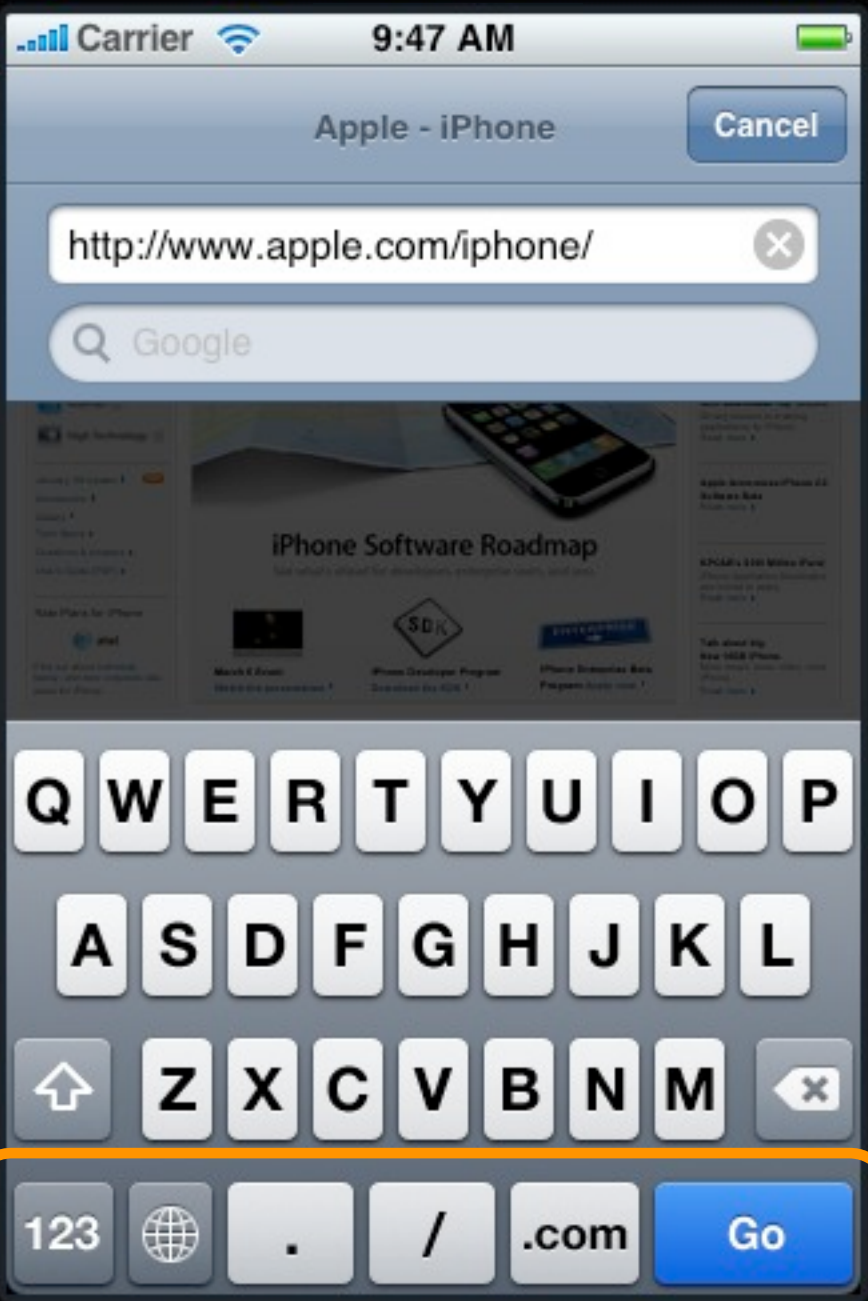
Return Key Type

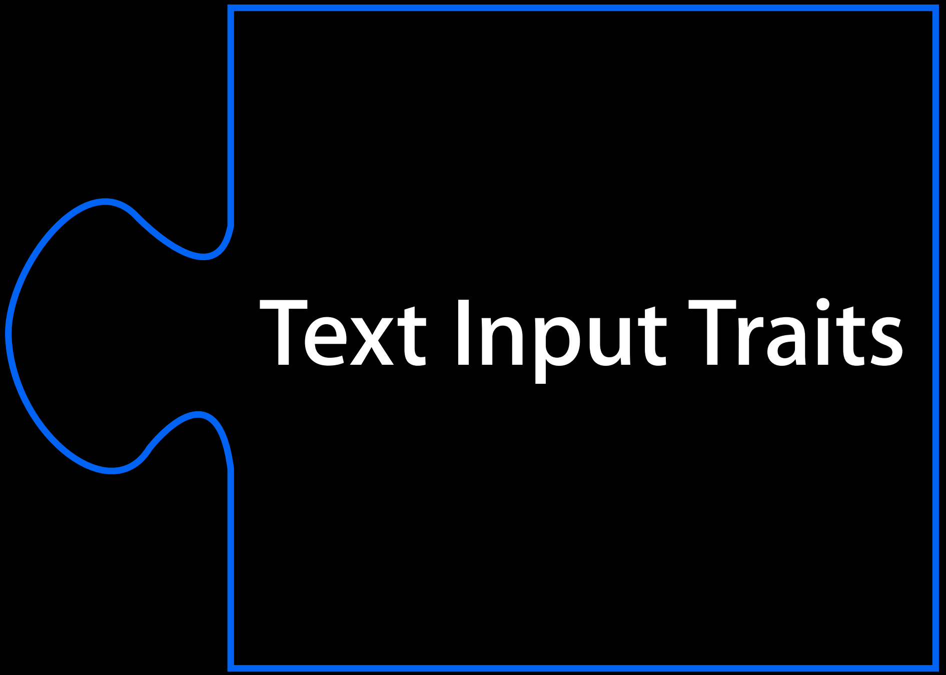
Return Key Autoenabling

Secure Text Entry

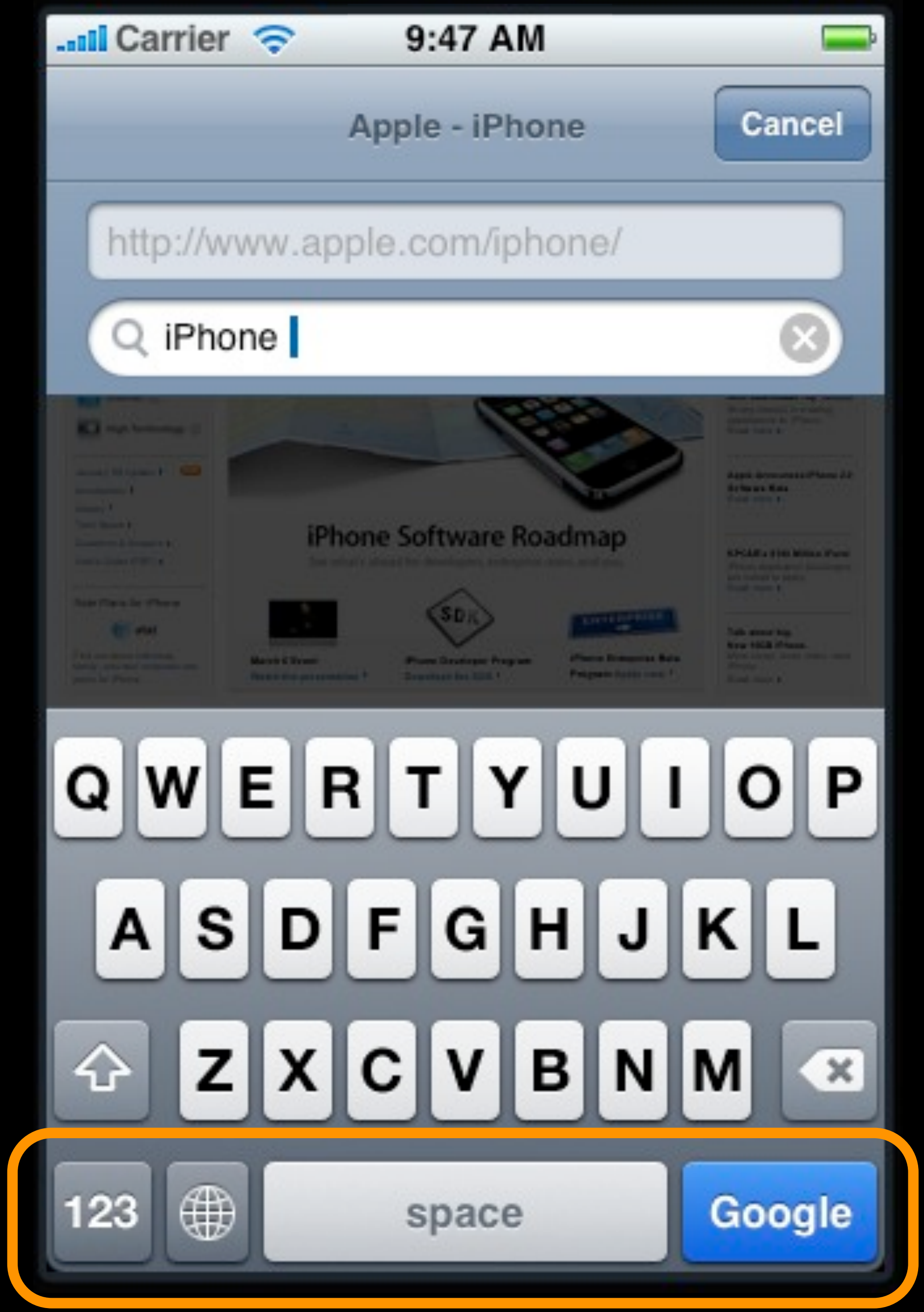
Text Input Traits

URL Keyboard
Go button





Default Keyboard
Google button



Text Containers

Text Input Traits

Delegates

Notifications

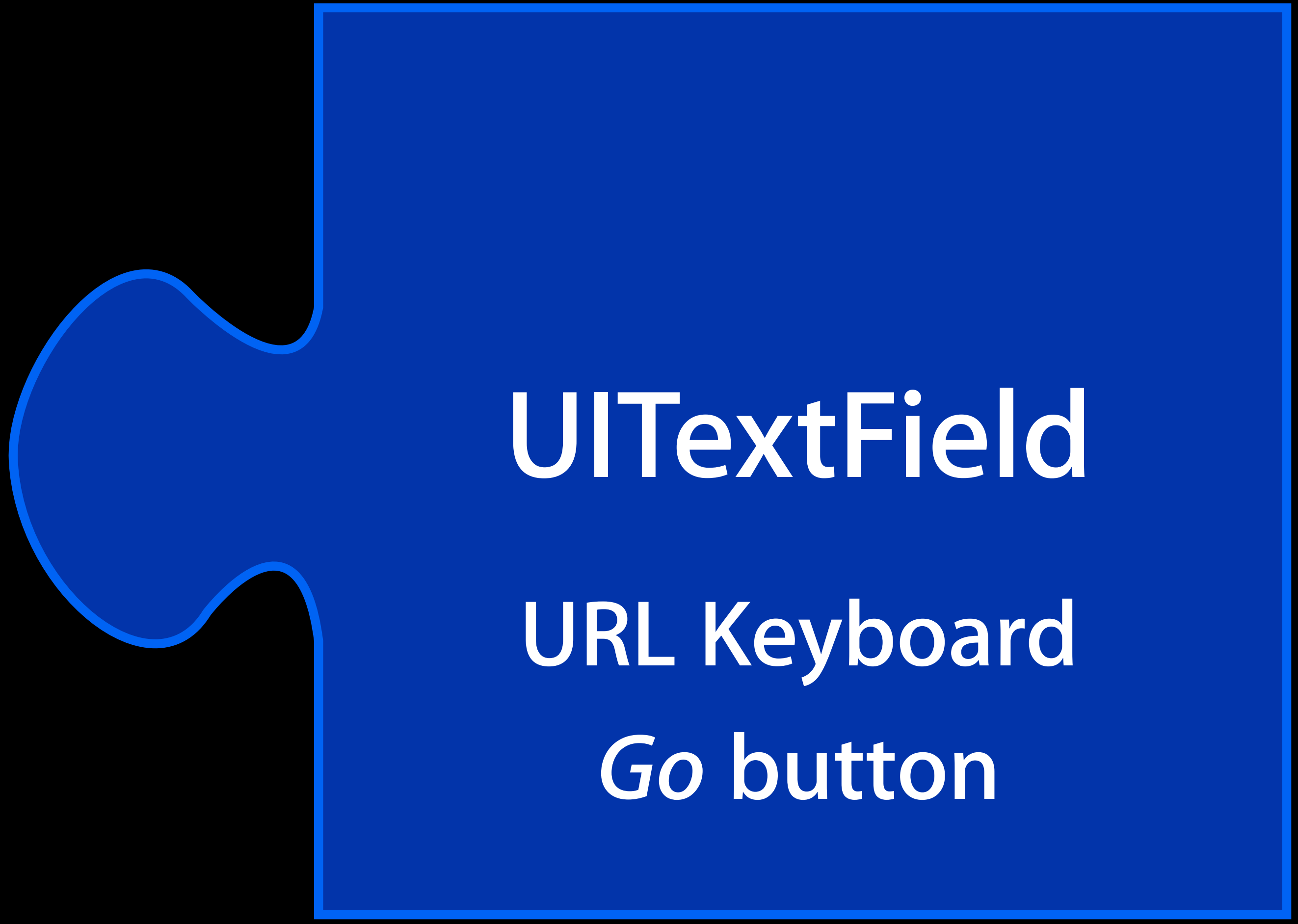
Methods

UITextField

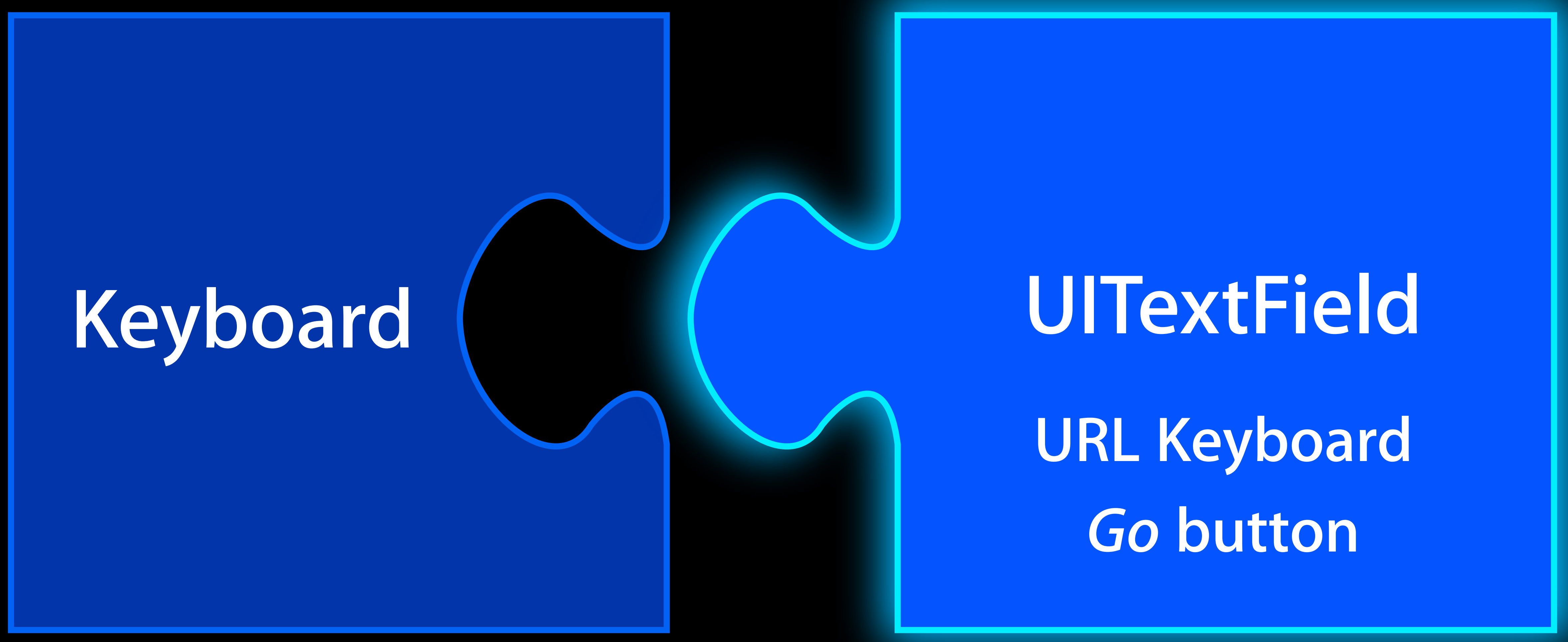
URL Keyboard

Go button

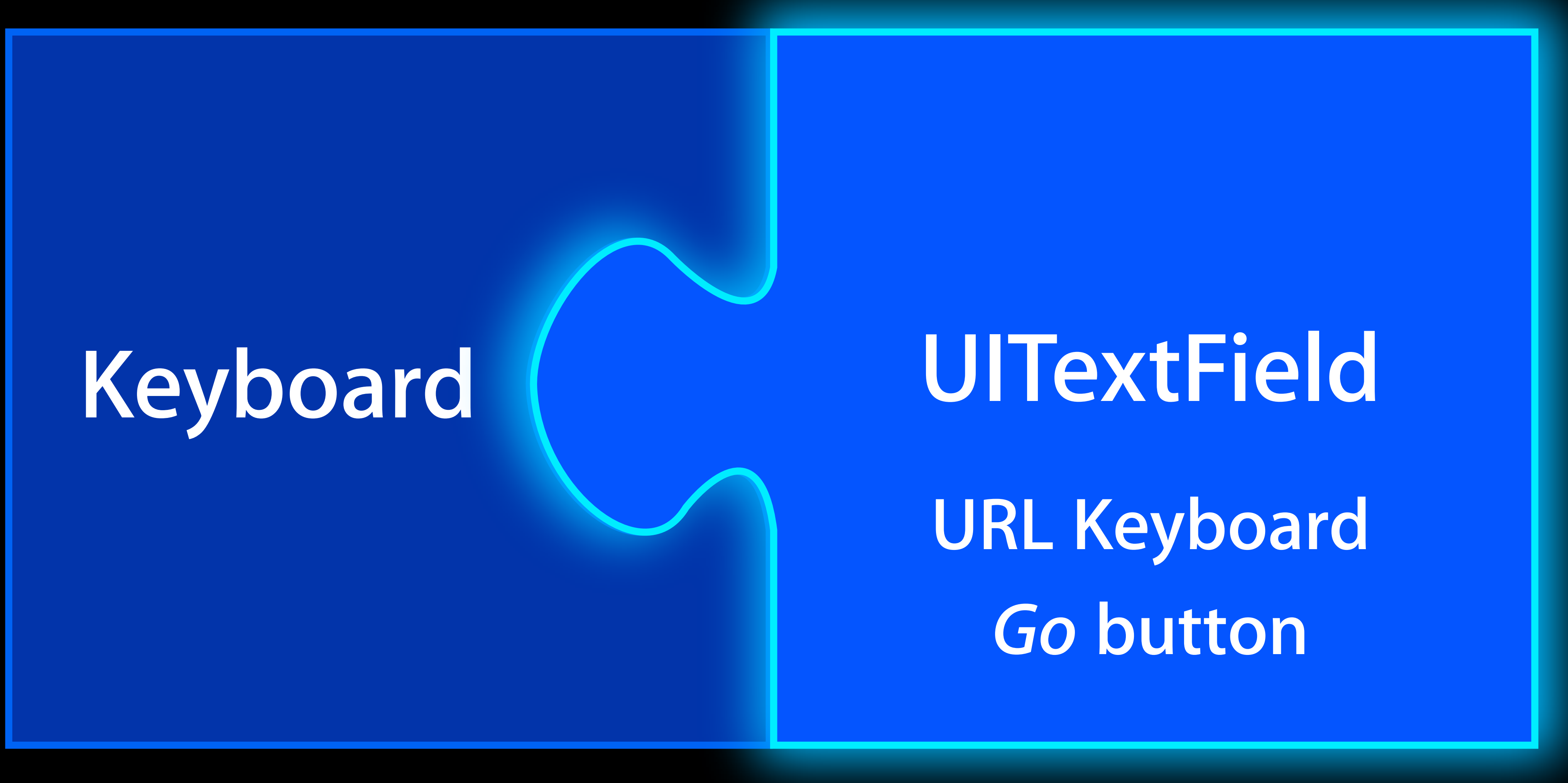
Design time



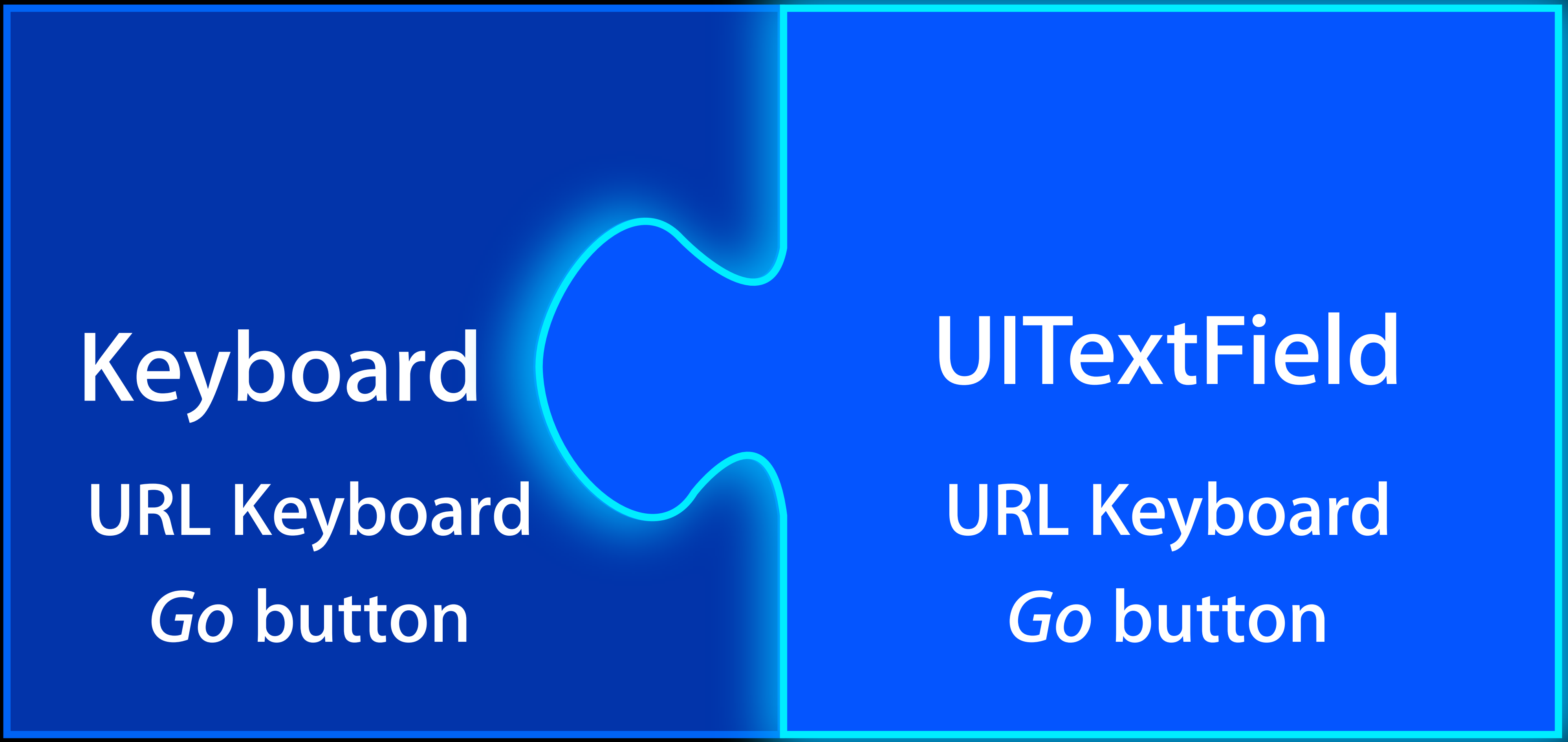
Run time



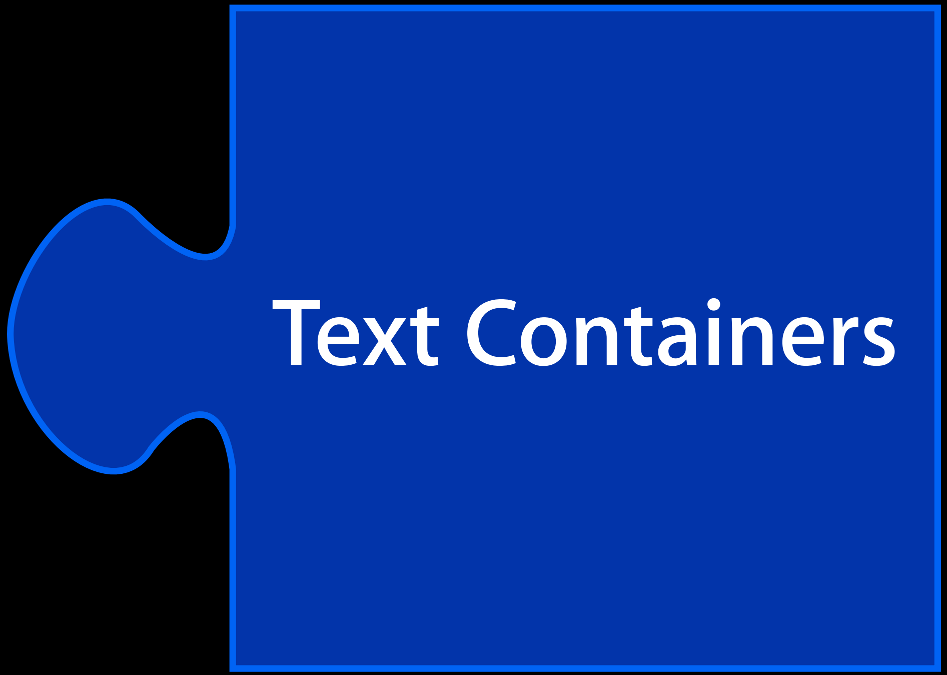
Become first responder



Become first responder



Keyboard adopts traits



UITextField

UITextView

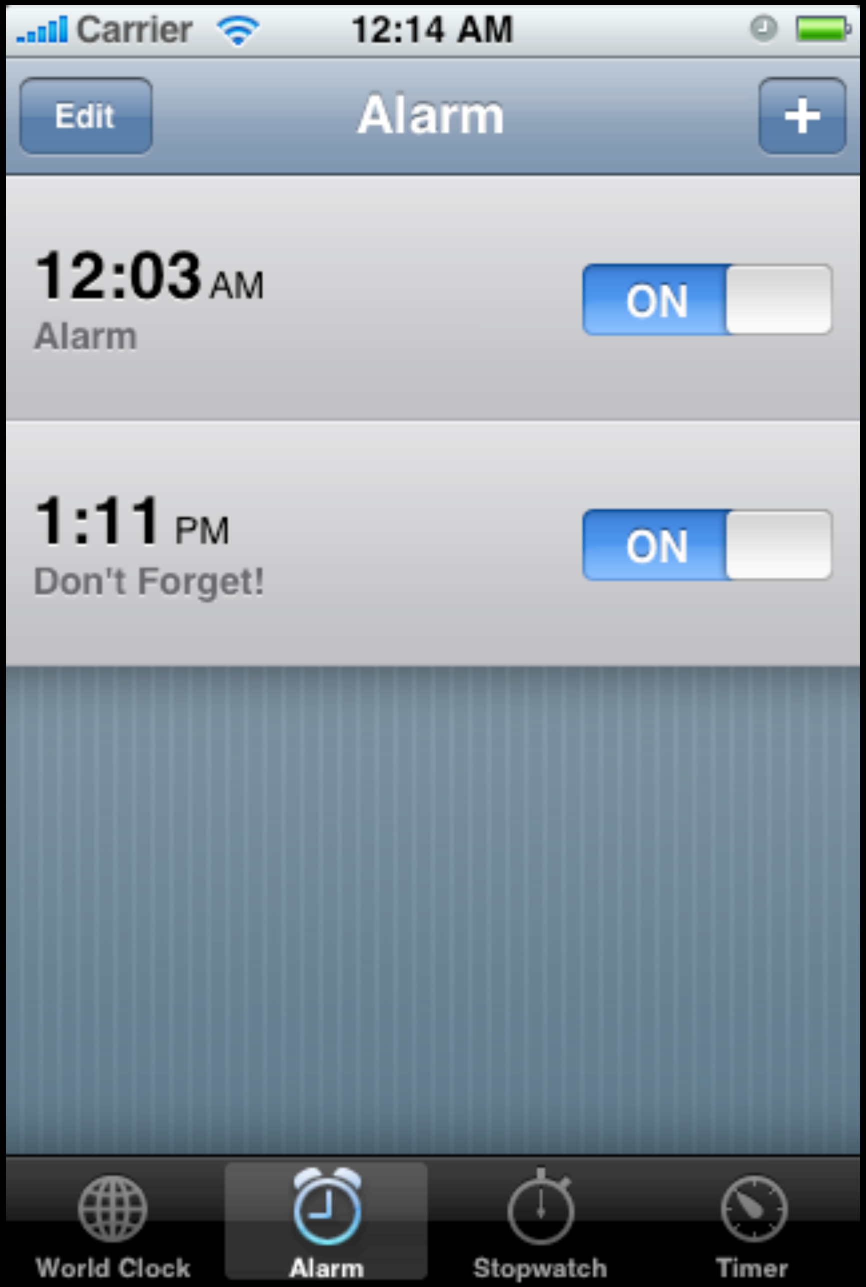
Web Forms

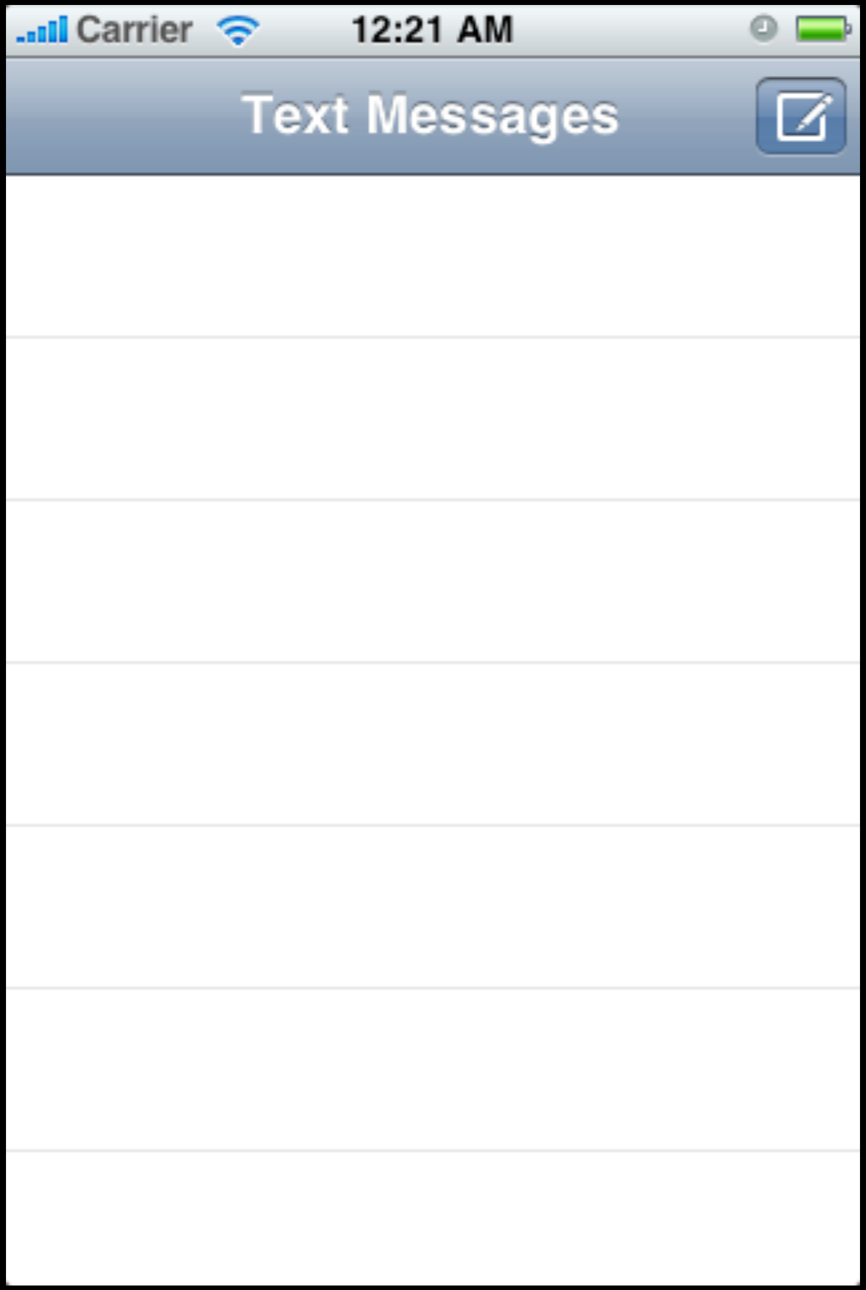
Demo:
Text Input

Presenting Content Modally

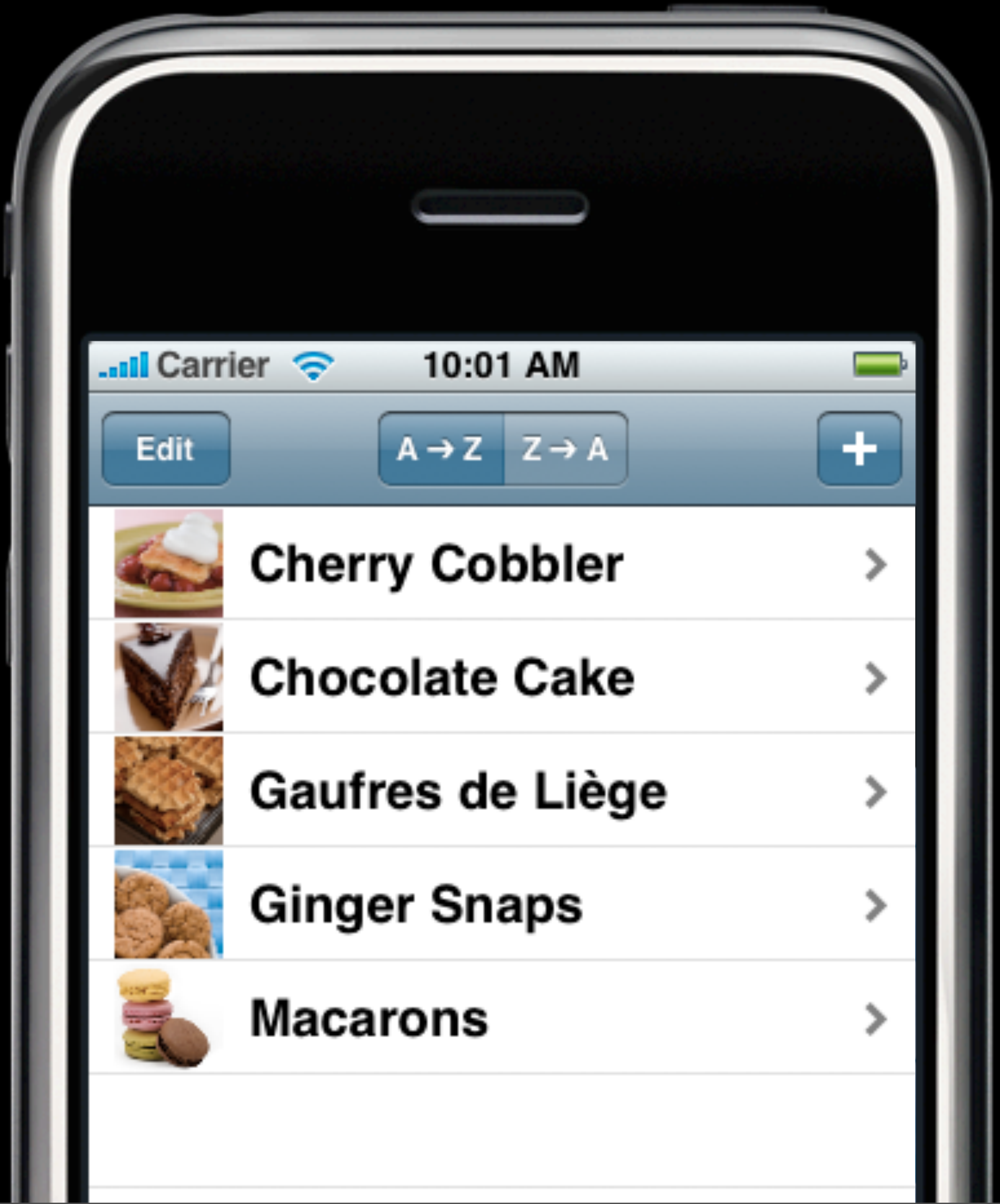
Presenting Content Modally

- For **adding** or **picking** data



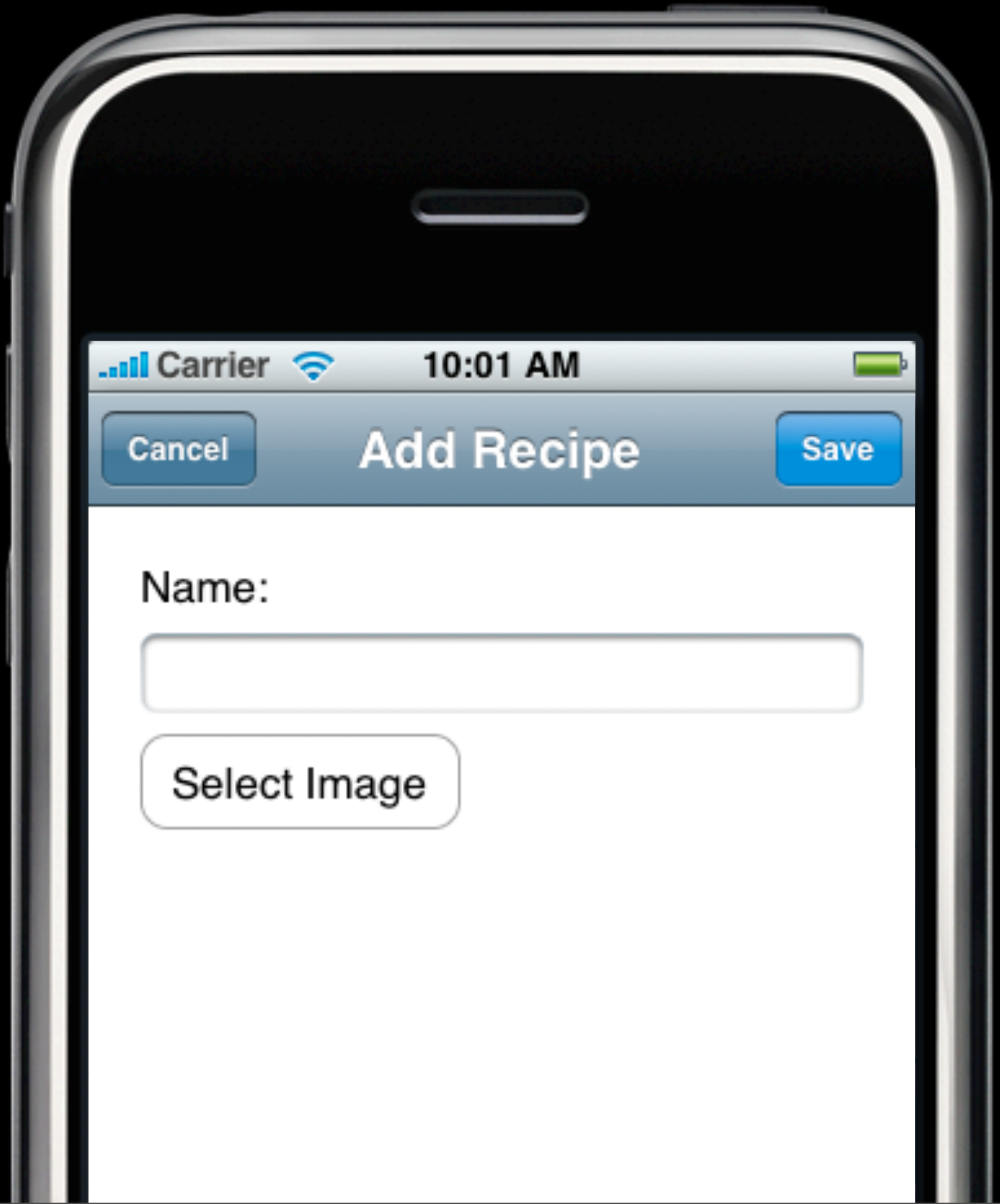


Presenting a View Controller

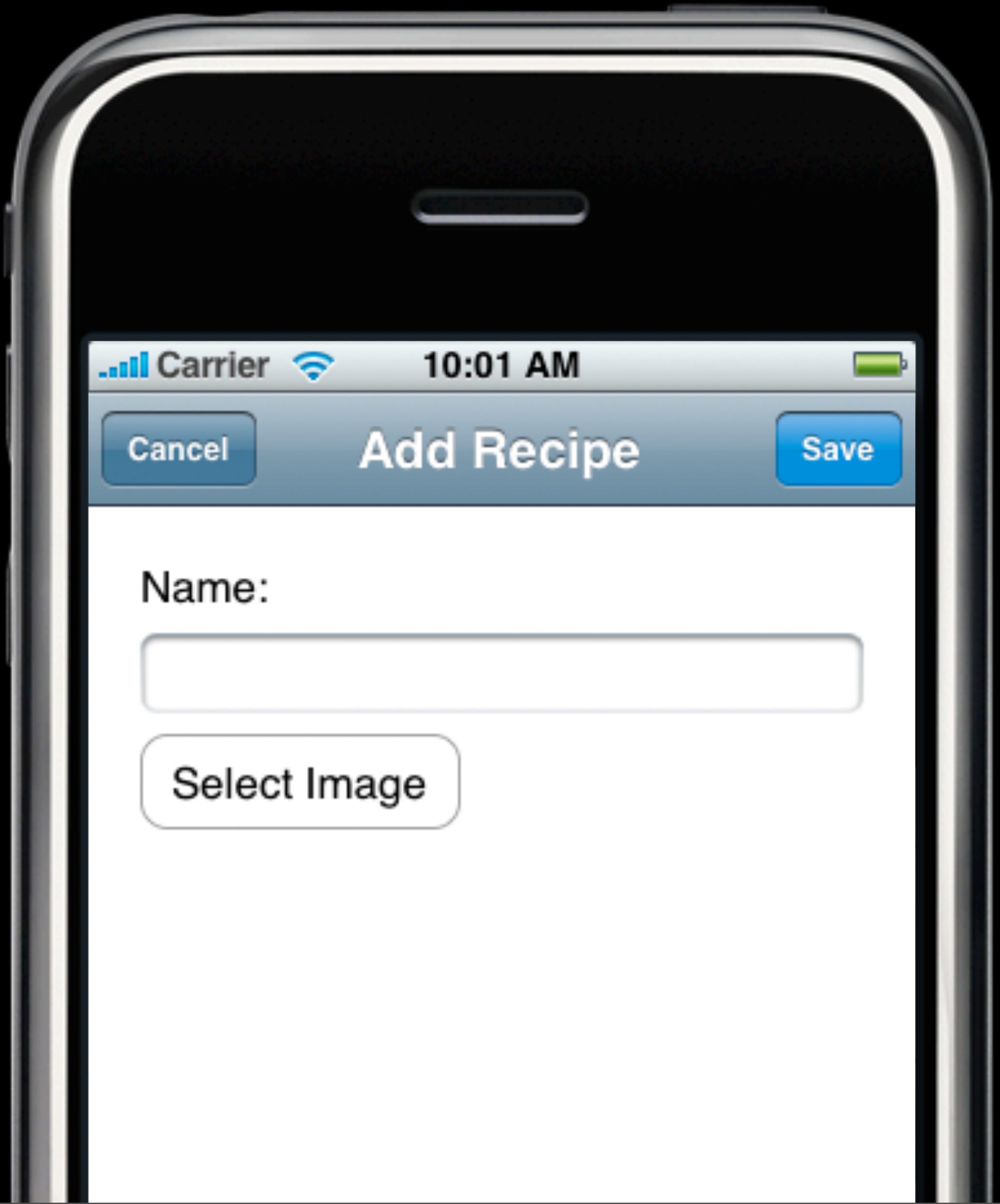


Presenting a View Controller

```
// Recipe list view controller
- (void)showAddRecipe {
   RecipeAddViewController *viewController = ...;
   [self presentModalViewController:viewController animated:YES];
}
```

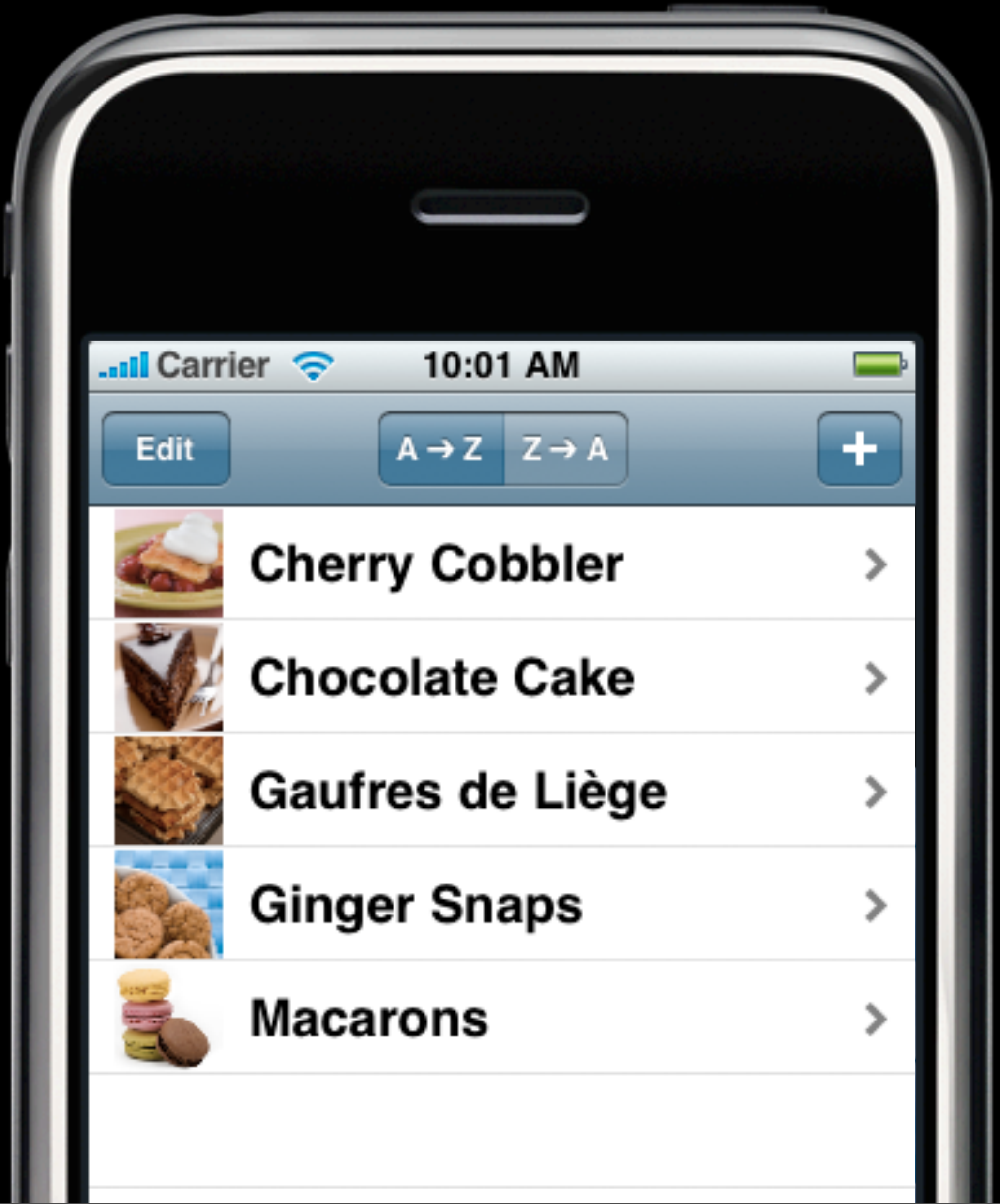


Dismissing a View Controller

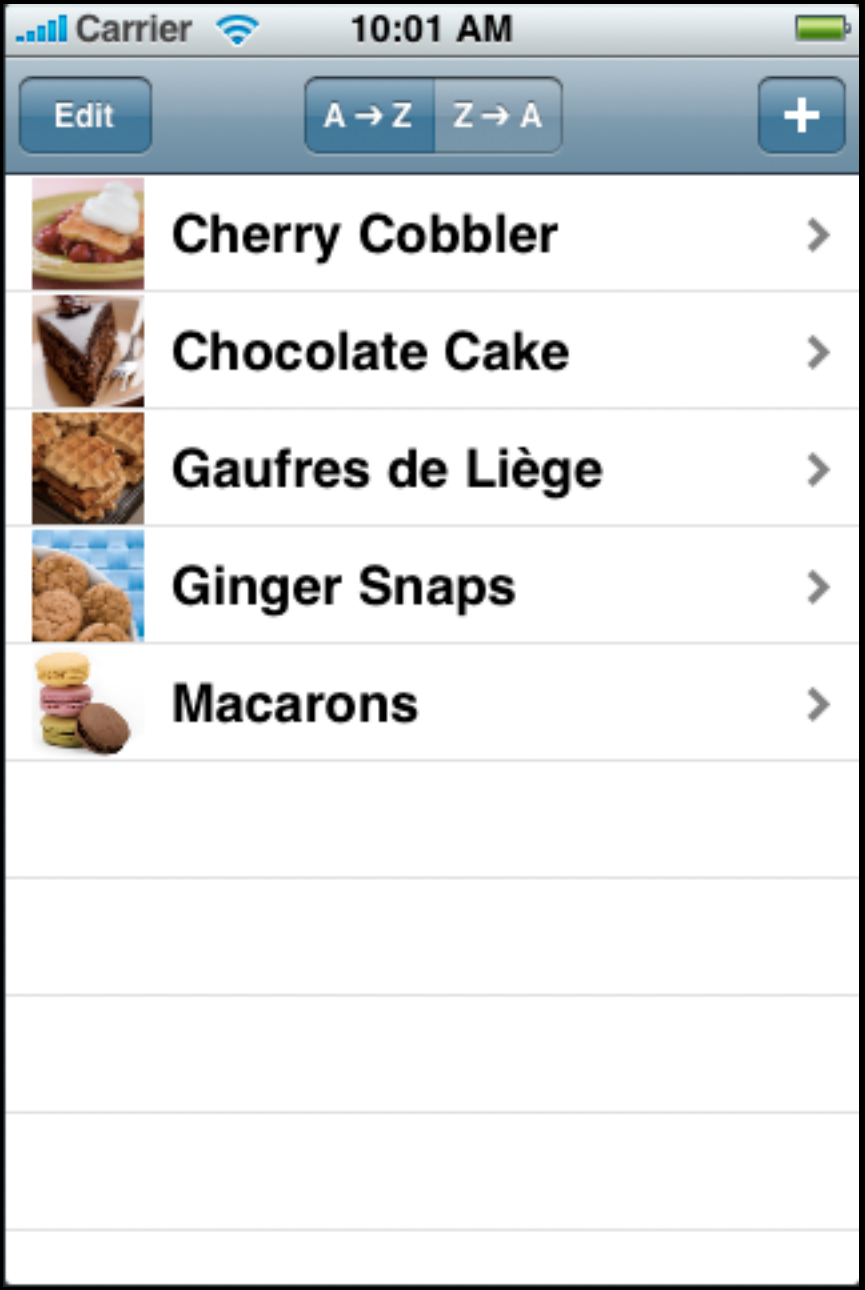


Dismissing a View Controller

```
// Recipe list view controller
- (void)didAddRecipe {
    [self dismissModalViewControllerAnimated:YES];
}
```

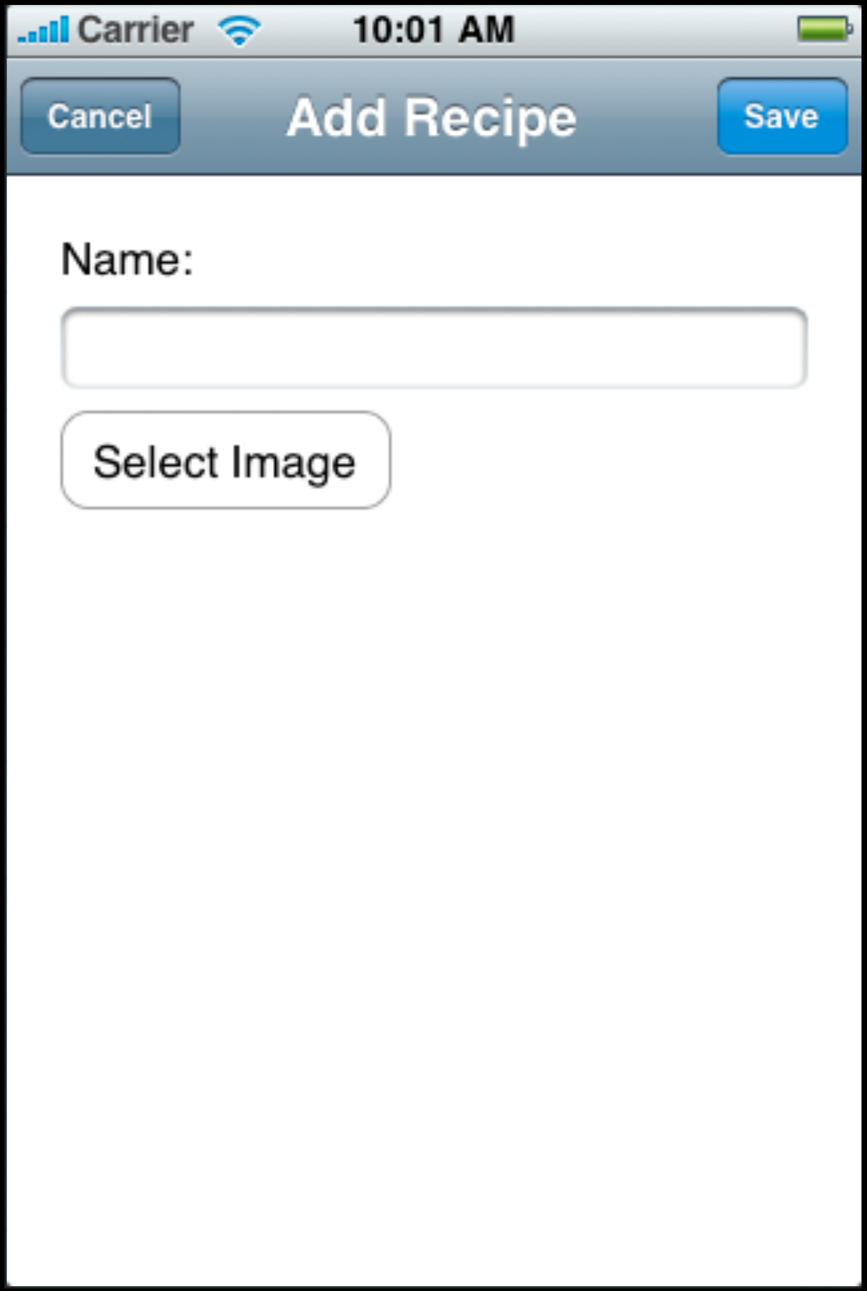


Separate Navigation Stacks

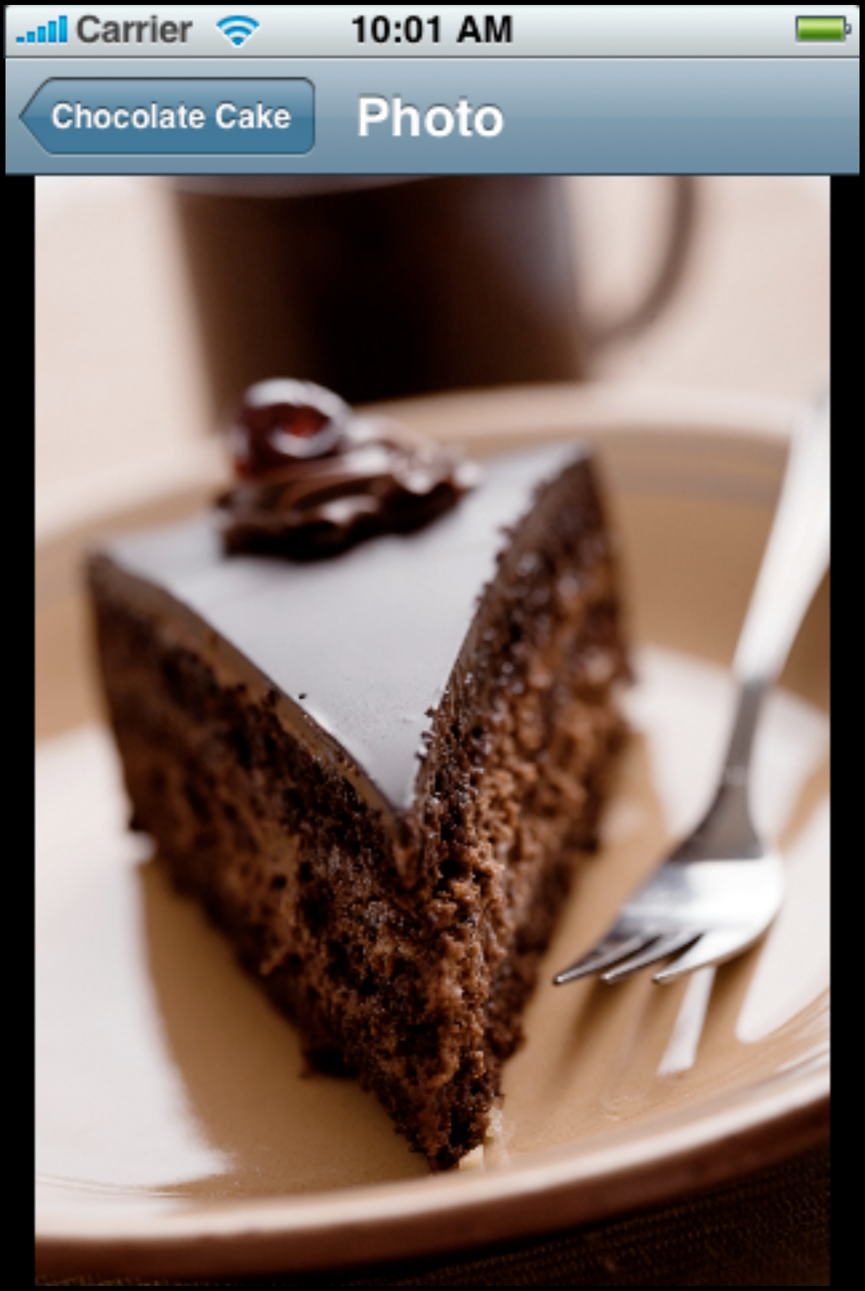


Cherry Cobbler
Chocolate Cake
Gaufres de Liège
Ginger Snaps
Macarons

Separate Navigation Stacks

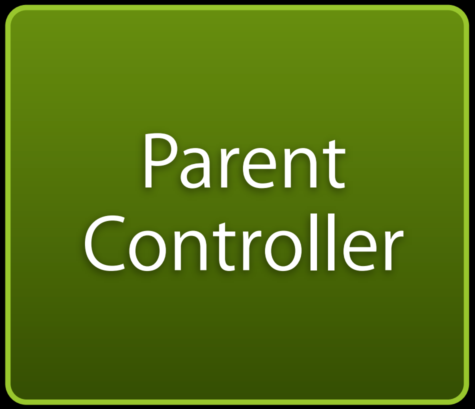


Separate Navigation Stacks



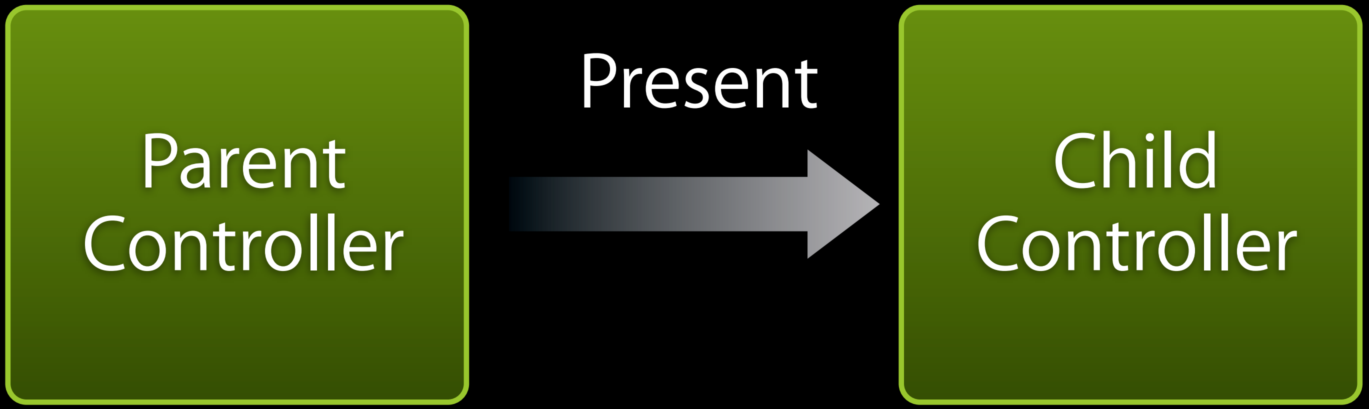
Dismissing a Modal View Controller

- Who should do it?
- Best practice is for the **same object** to call present and dismiss
- **Define delegate methods** for the presented controller
  - Tell the delegate when the presented controller is done
  - The delegate makes the call to dismiss



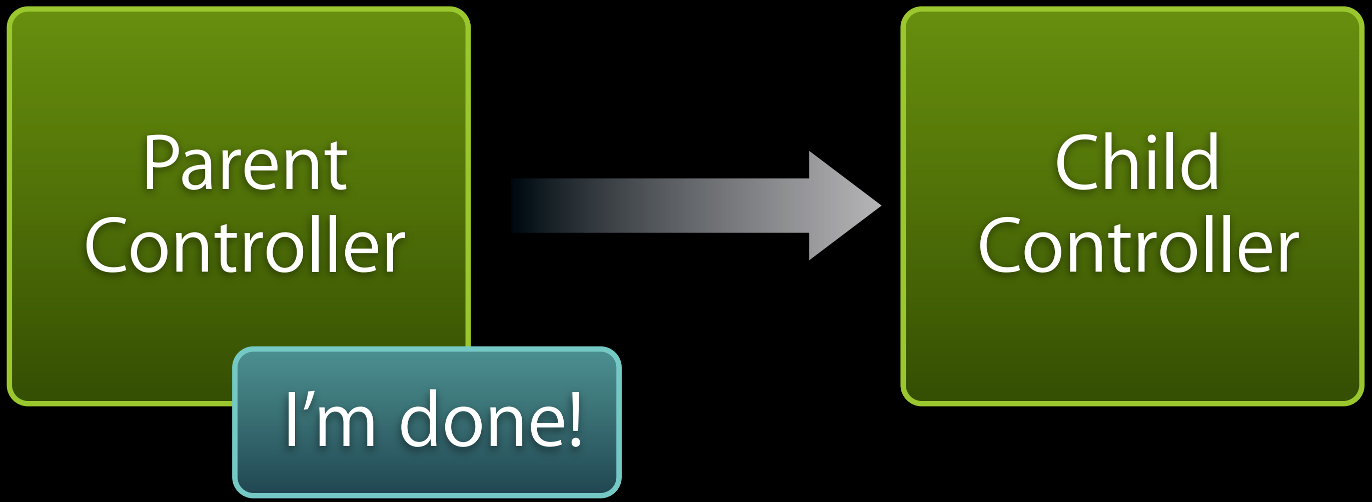
Dismissing a Modal View Controller

- Who should do it?
- Best practice is for the **same object** to call present and dismiss
- **Define delegate methods** for the presented controller
    - Tell the delegate when the presented controller is done
    - The delegate makes the call to dismiss



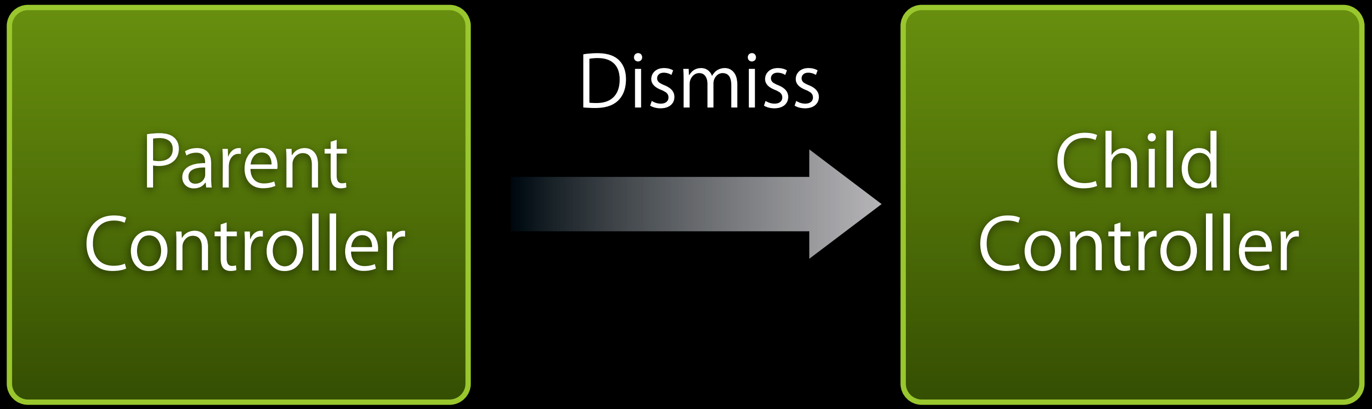
Dismissing a Modal View Controller

- Who should do it?
- Best practice is for the **same object** to call present and dismiss
- **Define delegate methods** for the presented controller
  - Tell the delegate when the presented controller is done
  - The delegate makes the call to dismiss



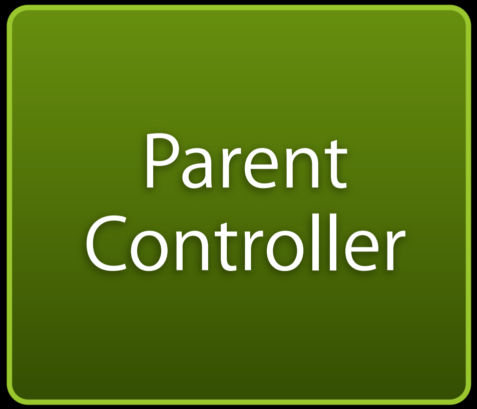
Dismissing a Modal View Controller

- Who should do it?
- Best practice is for the **same object** to call present and dismiss
- **Define delegate methods** for the presented controller
    - Tell the delegate when the presented controller is done
    - The delegate makes the call to dismiss



Dismissing a Modal View Controller

- Who should do it?
- Best practice is for the **same object** to call present and dismiss
- **Define delegate methods** for the presented controller
  - Tell the delegate when the presented controller is done
  - The delegate makes the call to dismiss



Demo:
Presenting Content Modally

Presence - Part 3

Goals for Presence 3

- Avoid expensive work on the main thread
  - Use background threads to **keep UI responsive**
  - Abstract thread lifecycle with NSOperation & NSOperationQueue
- Allow the user to update their own status
  - Present a view controller modally
  - Customize text input traits on a UITextField
  - Use a delegate callback when finished

Questions?