

CS193P - Lecture 11

iPhone Application Development

Text Input
Presenting Content Modally

テキスト入力
ダイアログ的なコンテンツ表示

Announcements

- Paparazzi 3 assignment is due Wednesday 2/17
- This Friday's extra session will feature Evan Doll

Today's Topics

- Threading Wrap-Up
- iPhone Keyboards
- Customizing Text Input
- Presenting Content Modally

(これは省略)

キーボード

テキスト入力のカスタマイズ

ダイアログ的(一時的)な
コンテンツ表示

NSOperation

- Abstract superclass
- Manages thread creation and lifecycle
- Encapsulate a **unit of work** in an object
- Specify priorities and dependencies

(省略)

Creating an NSOperation Subclass

- Define a **custom init method**

```
- (id)initWithSomeObject:(id)someObject
{
    self = [super init];
    if (self) {
        self.someObject = someObject;
    }
    return self;
}
```

- **Override -main method** to do work

```
- (void)main
{
    [someObject doLotsOfTimeConsumingWork];
}
```

(省略)

NSOperationQueue

- Operations are typically scheduled by **adding to a queue**
- Choose a maximum number of concurrent operations
- Queue runs operations based on priority and dependencies

Using an NSInvocationOperation

- Concrete subclass of NSOperation
- For lightweight tasks where creating a subclass is overkill

```
- (void)someAction:(id)sender
{
    NSInvocationOperation *operation =
        [[NSInvocationOperation alloc] initWithTarget:self
        selector:@selector(doWork:)
        object:someObject];

    [queue addObject:operation];

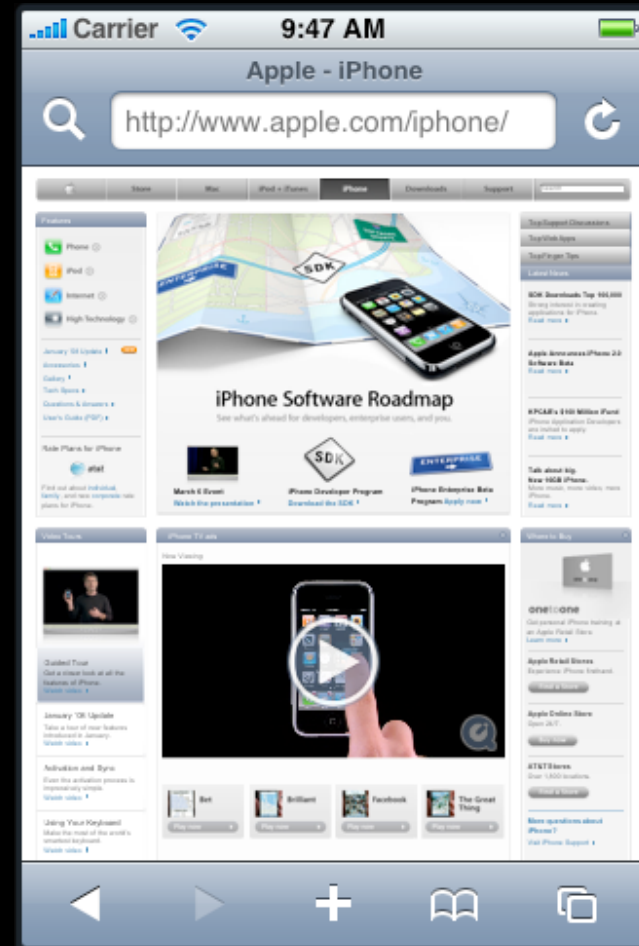
    [operation release];
}
```

iPhone Keyboards

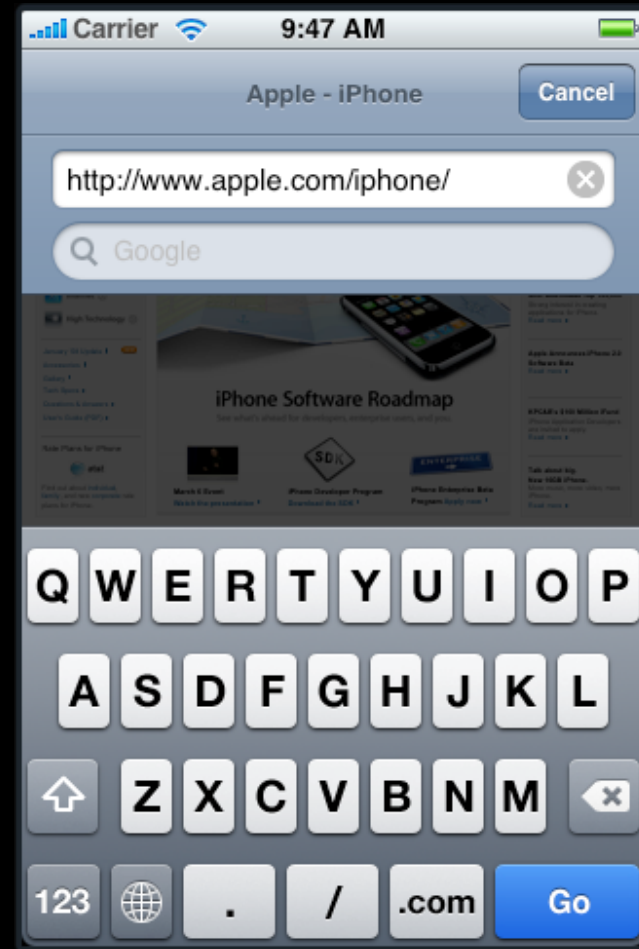
iPhone キーボード

Virtual keyboard Appears when needed

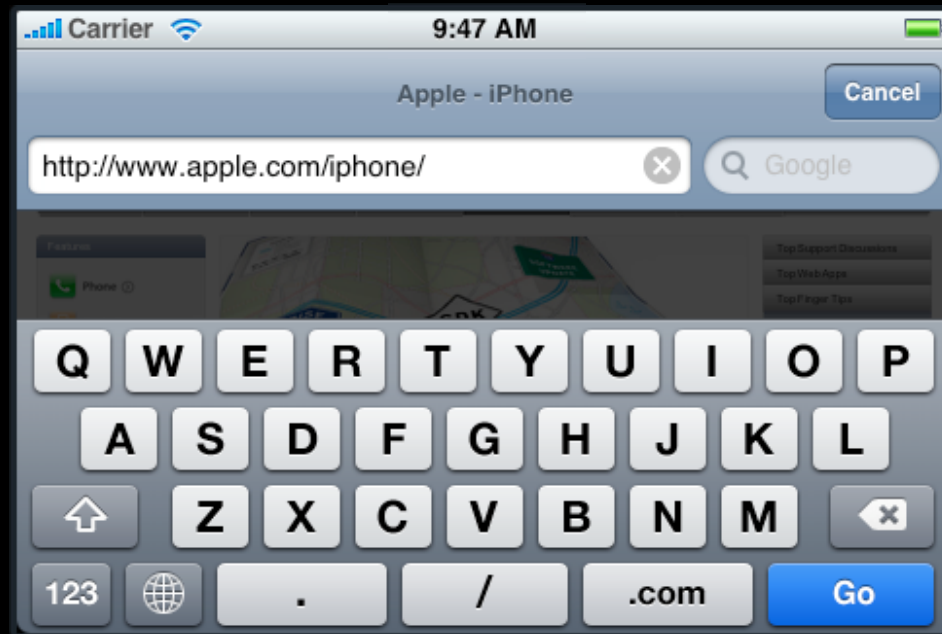
仮想キーボード
必要に応じて現われる



Virtual keyboard Appears when needed







Portrait and Landscape

Simple selection model Text loupe/magnifier

シンプルな選択モデル
テキスト用のルーペ/拡大鏡

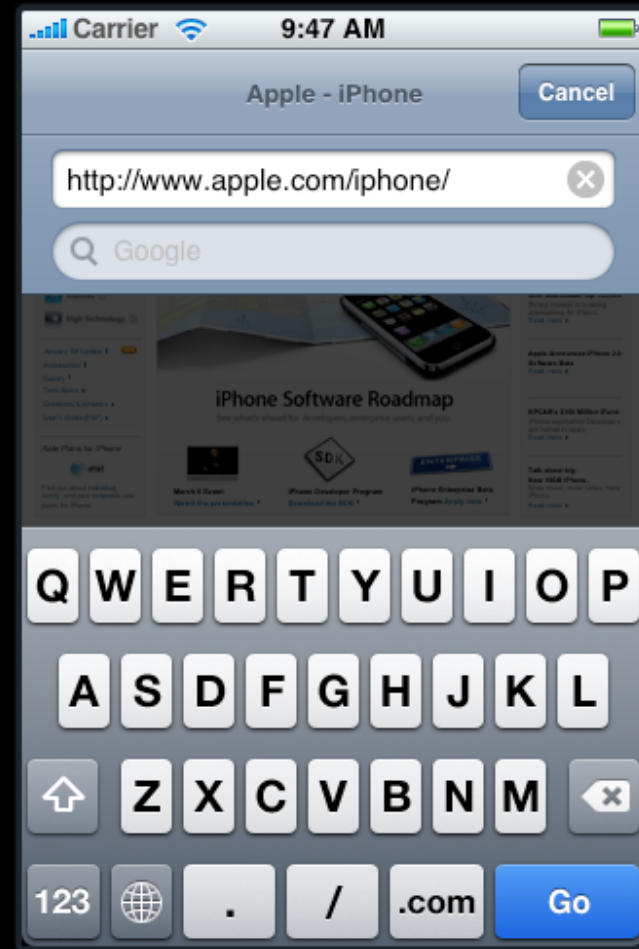


Many keyboard types Adapted to task

たくさんのキーボードタイプ
タスクに適応

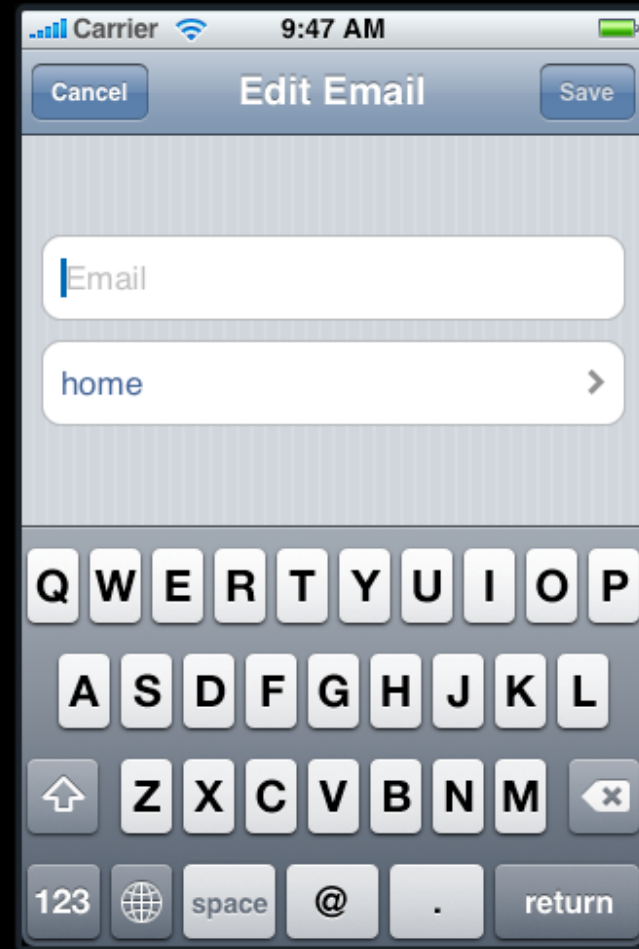


Many keyboard types Adapted to task



URL 入力用

Many keyboard types Adapted to task



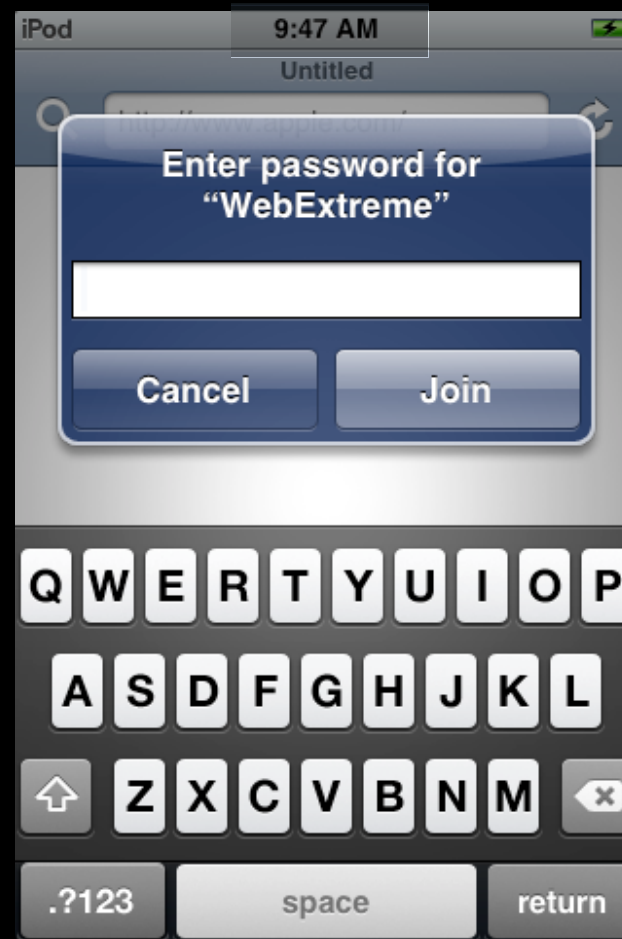
メアド入力用

Many keyboard types
Adapted to task



電話番号入力用

Many keyboard types
Adapted to task



パスワード入力用

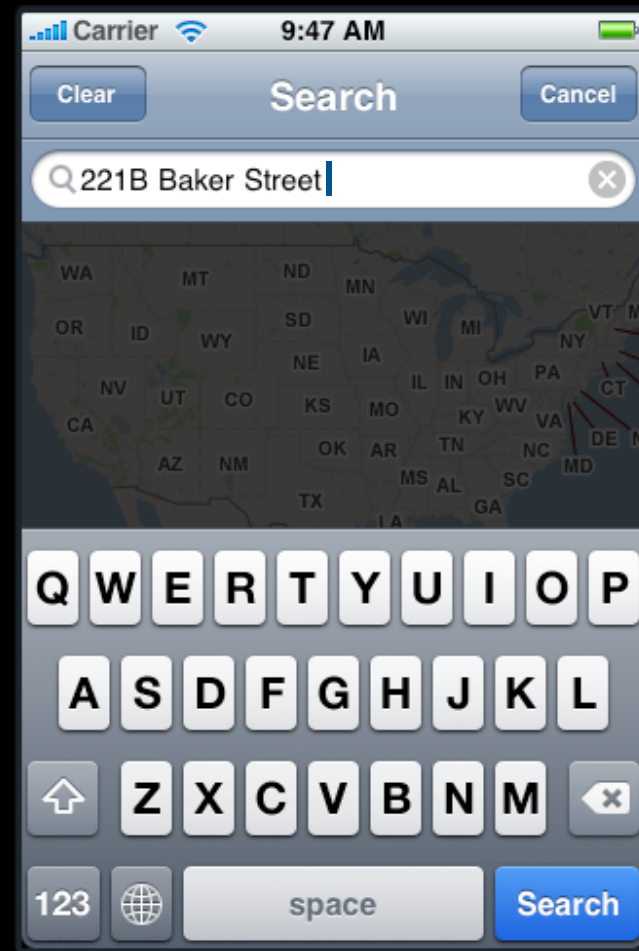
Many keyboard types
Adapted to task



パスコード (4桁数字)
入力用

Single line editing

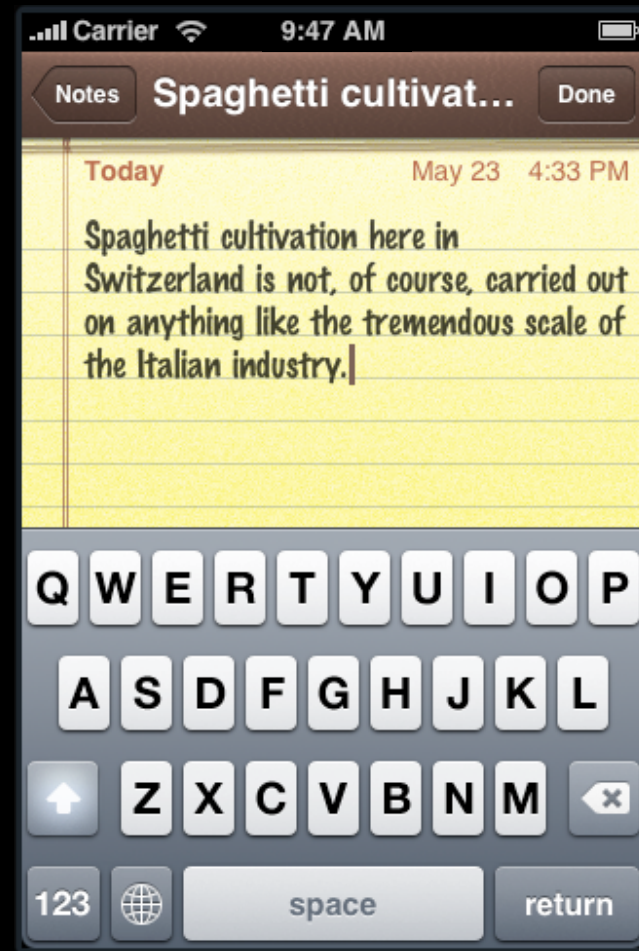
単一行での編集



Multi-line editing

複数行にわたる編集

40カ国語の
入力辞書サポート

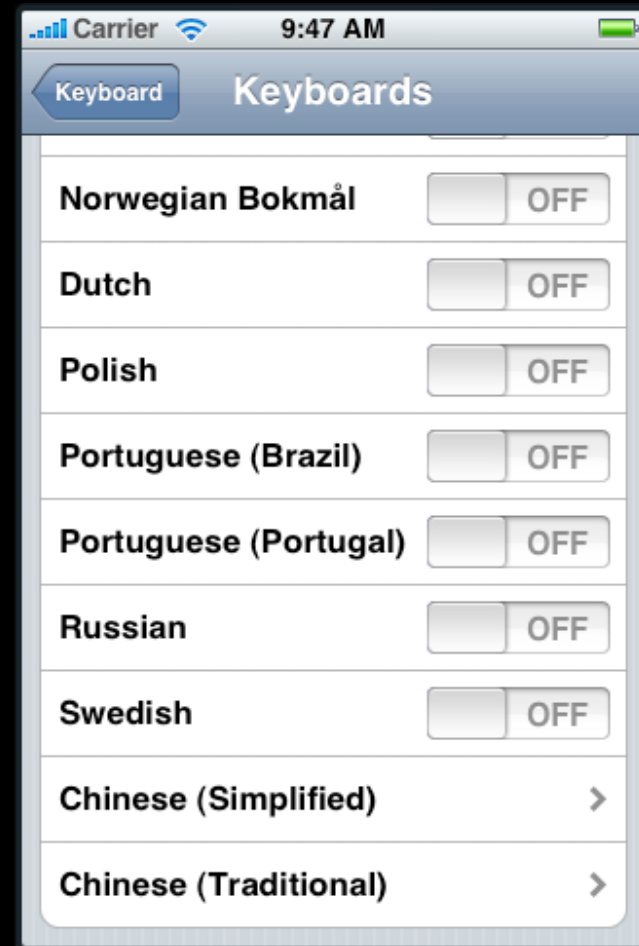


40

Languages

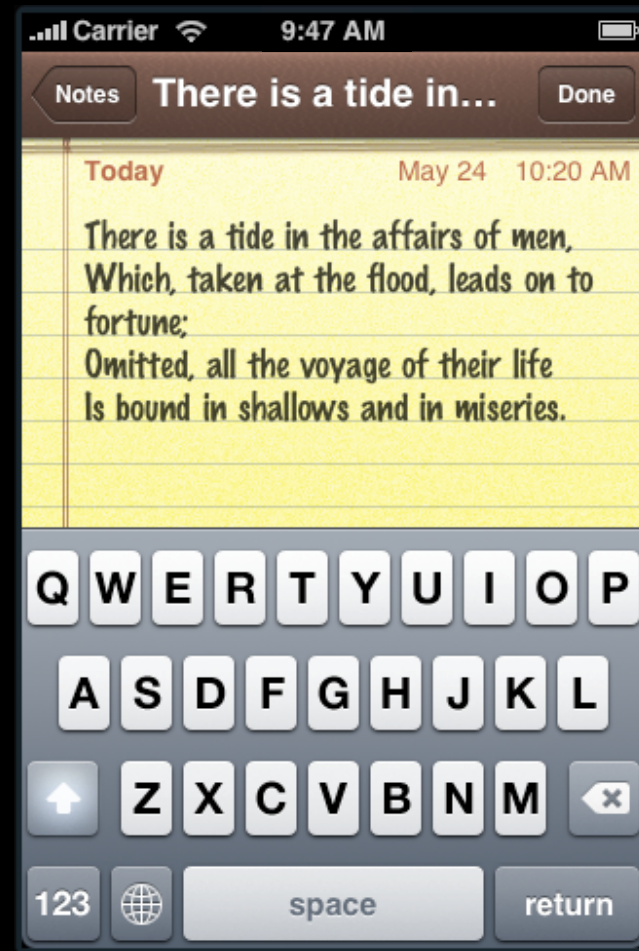
Full dictionary support

40カ国語について
入力辞書をサポート



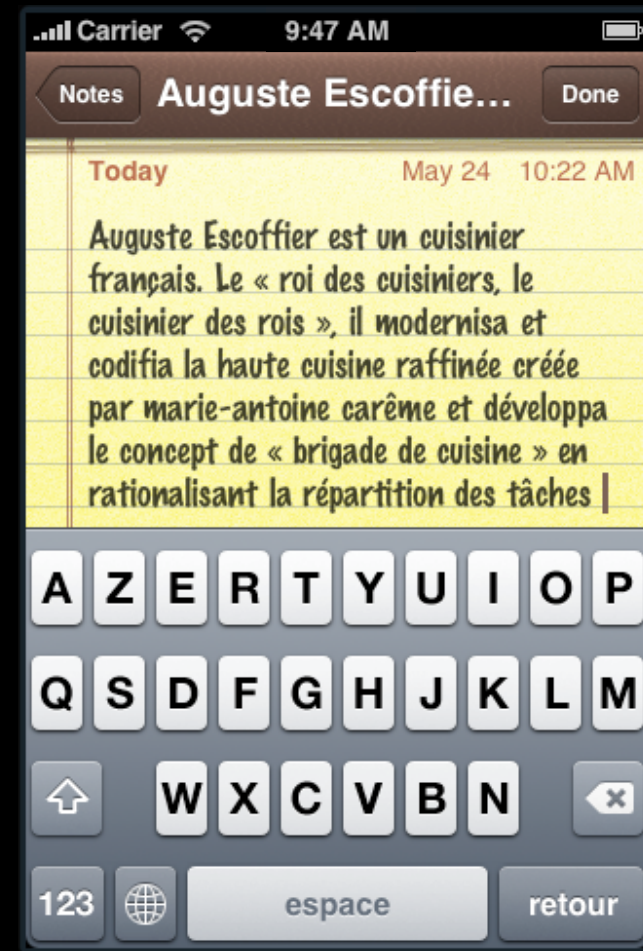
English

英語



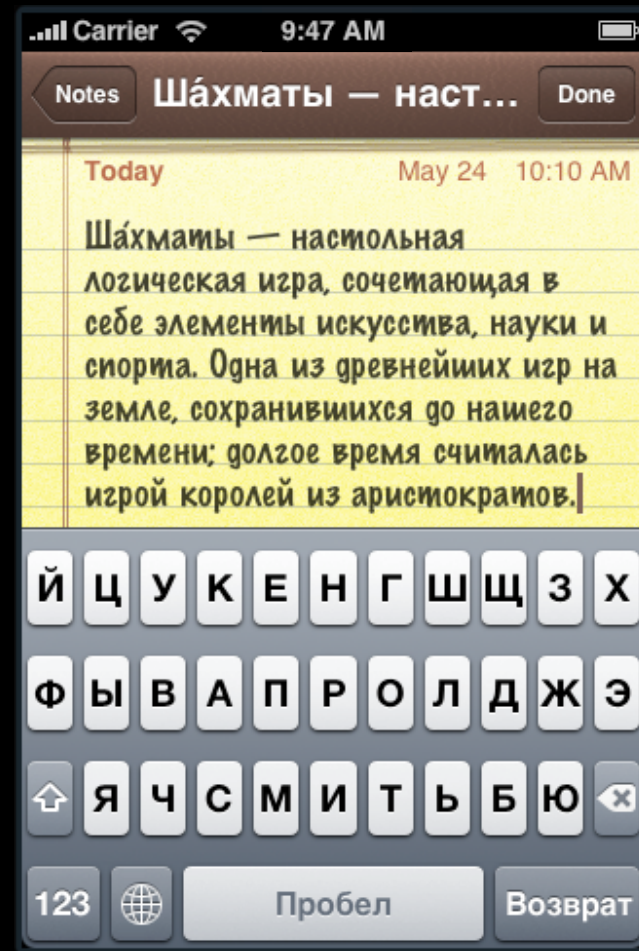
French

フランス語



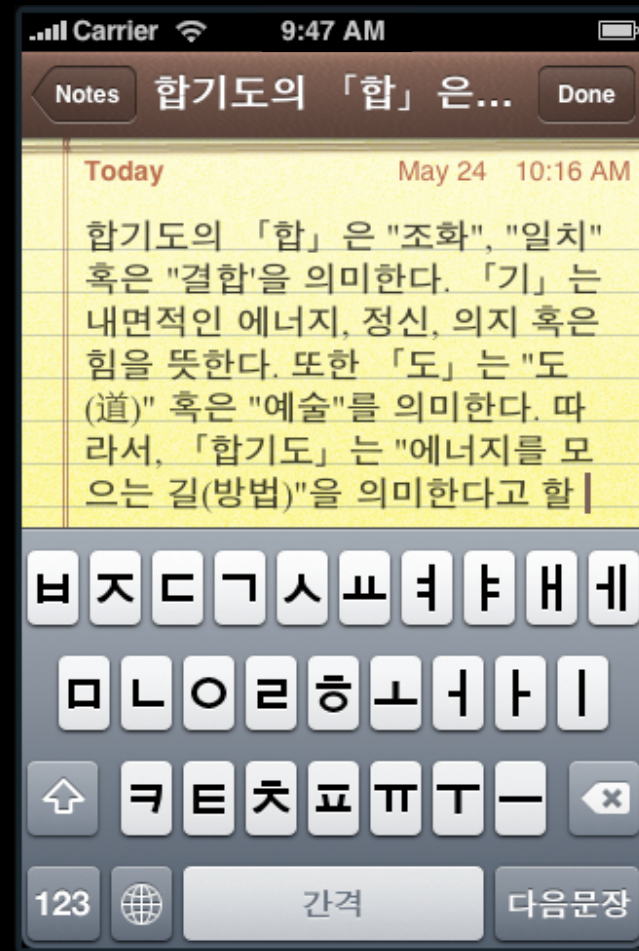
Russian

ロシア語



Korean

韓国/朝鮮語



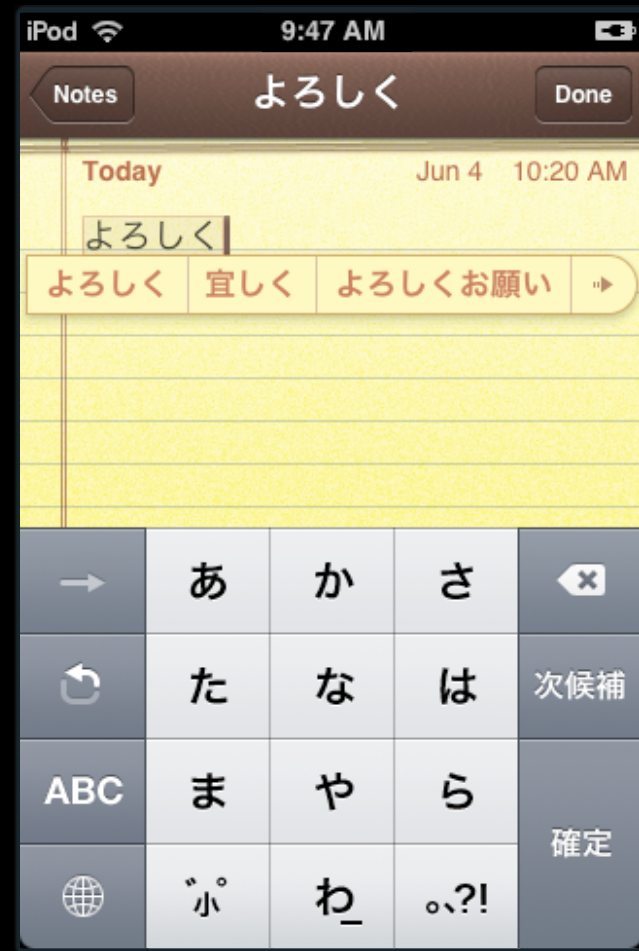
Japanese Romaji

日本語（ローマ字入力）



Japanese Kana

日本語（テンキー/カナ入力）



Chinese Pinyin

中国語（拼音字母による入力）



Chinese Handwriting

Simplified
Traditional

中国語（手書き入力）
簡体字（电气）
繁体字（電氣）



Customizing Text Input

テキスト入力のカスタマイズ



Text Containers

UITextView - - - 複数行のテキスト
UITextField - - - 単一行のテキスト

Text Containers

Delegates
Notifications
Methods

コントローラと
情報をやりとりする
従来の3つの方法

UITextView - - - 複数行のテキスト
UITextField - - - 単一行のテキスト

Text Containers

Text Input Traits

テキスト入力特性

UITextView - - - 複数行のテキスト

UITextField - - - 単一行のテキスト

テキスト入力特性 ----- プロトコル



Text Input Traits

Protocol

UITextField

UITextView

テキスト入力特性

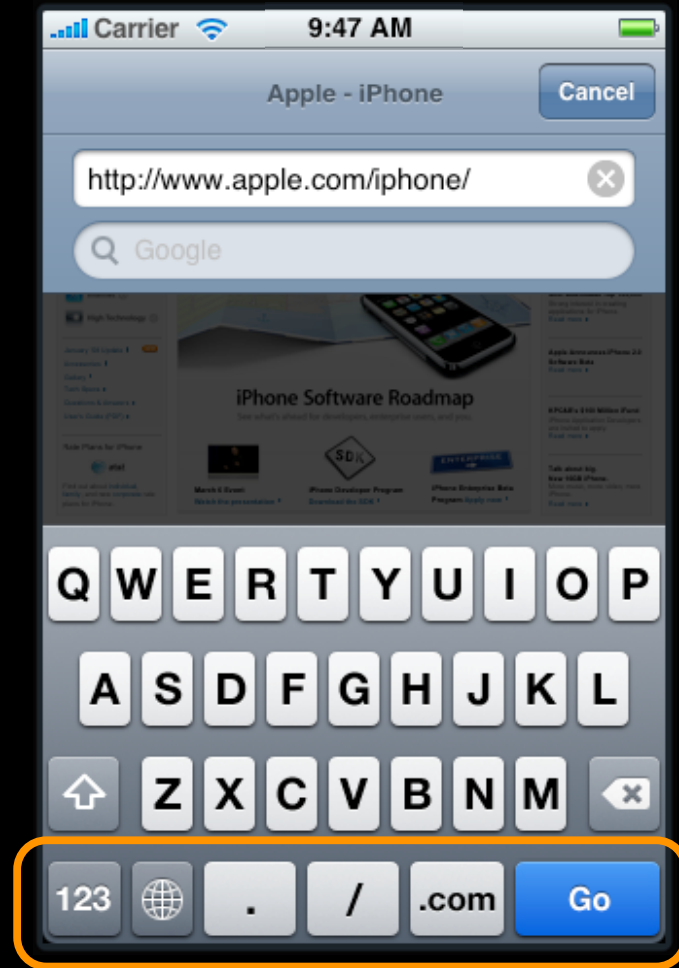
Text Input Traits

Autocapitalization	大文字化
Autocorrection	スペル訂正
Keyboard Type	型 (タイプ)
Keyboard Appearance	外見
Return Key Type	リターンキー
Return Key Autoenabling	
Secure Text Entry	何か入力されたらリターンキーを有効にする

テキスト入力特性

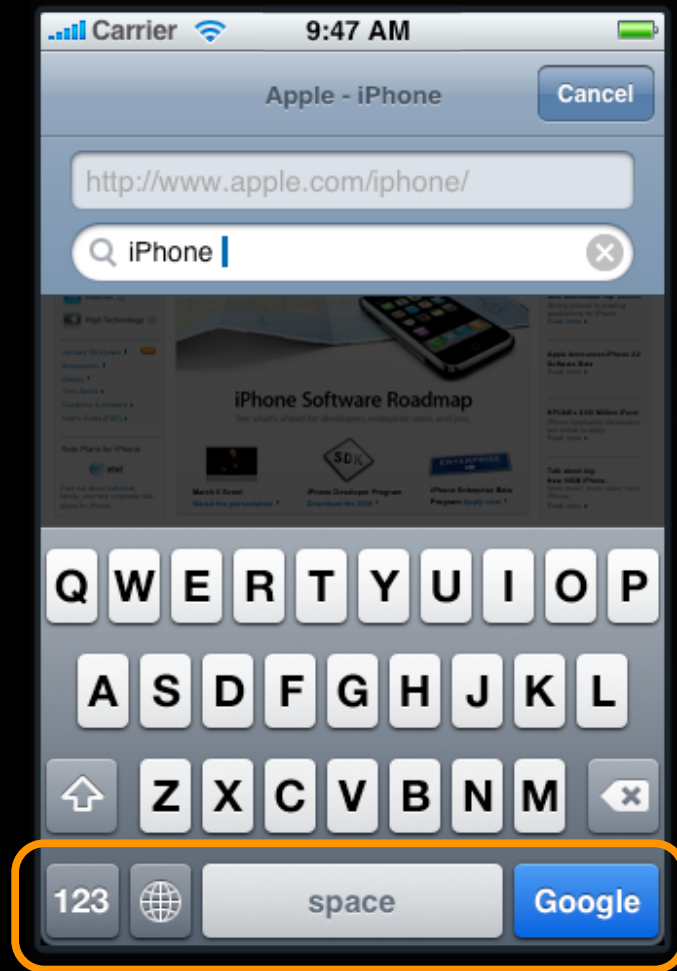
Text Input Traits

キーボードのタイプ
URL Keyboard
Go button
リターンキーのタイプ



Text Input Traits

標準タイプ
Default Keyboard
Google button
ぐるぐるボタン



Text Containers

Text Input Traits

Delegates

Notifications

Methods



Text Containers

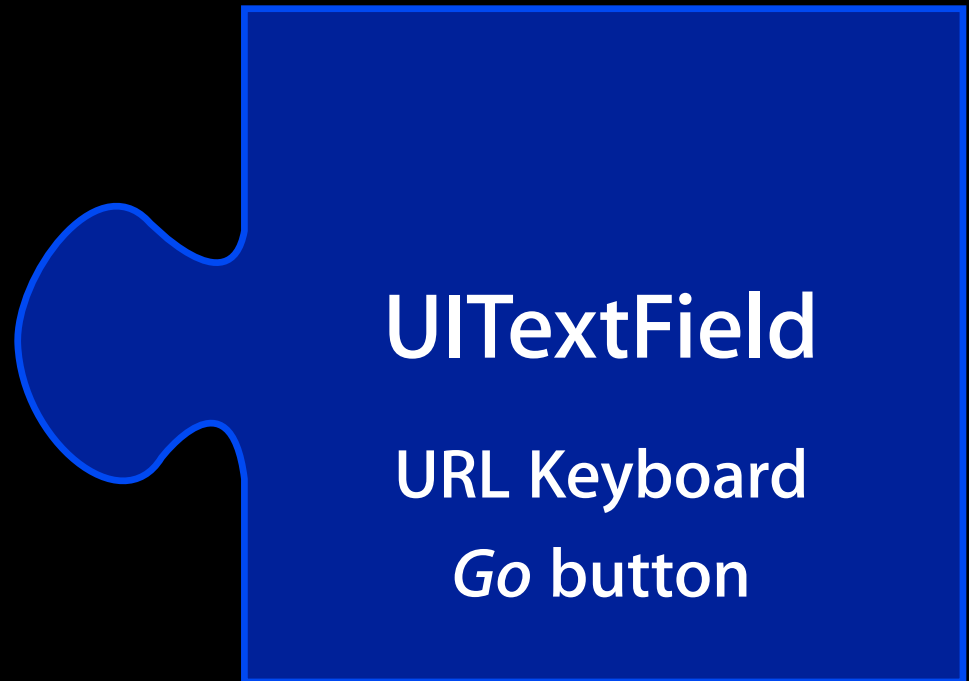
UITextField
UITextView
など



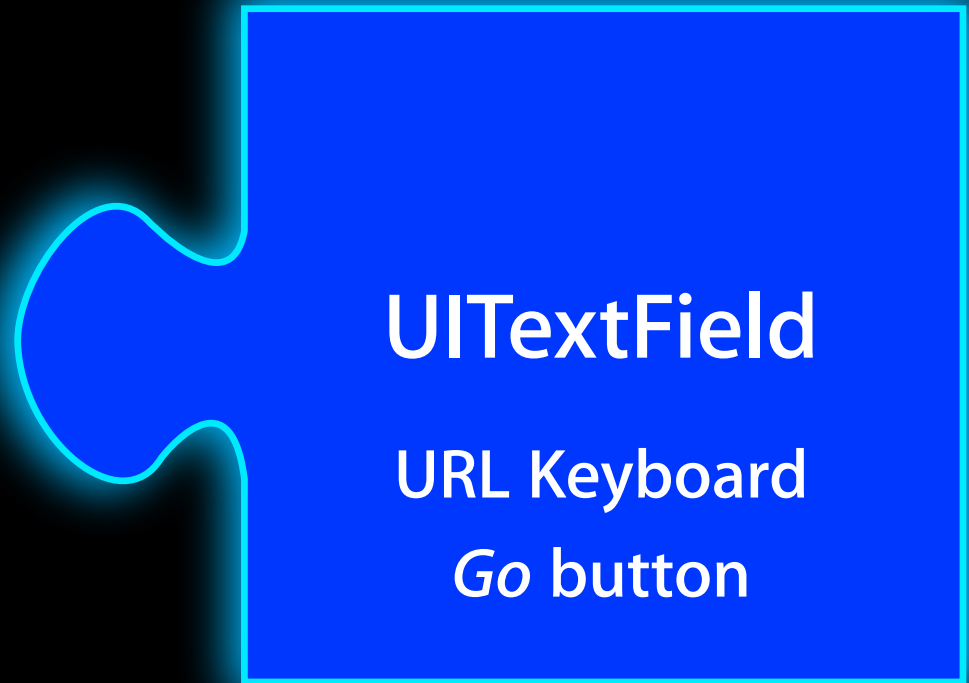
UITextField
URL Keyboard
Go button

入力特性を
デザイン

Design time



Run time
実行時は...



Become first responder
UITextField がユーザ応答の責任をもつ



Keyboard

UITextField

URL Keyboard

Go button

キーボードが現われる

Become first responder



Keyboard

UITextField

URL Keyboard

Go button

Become first responder



キーボードに UITextField が接続される

Become first responder



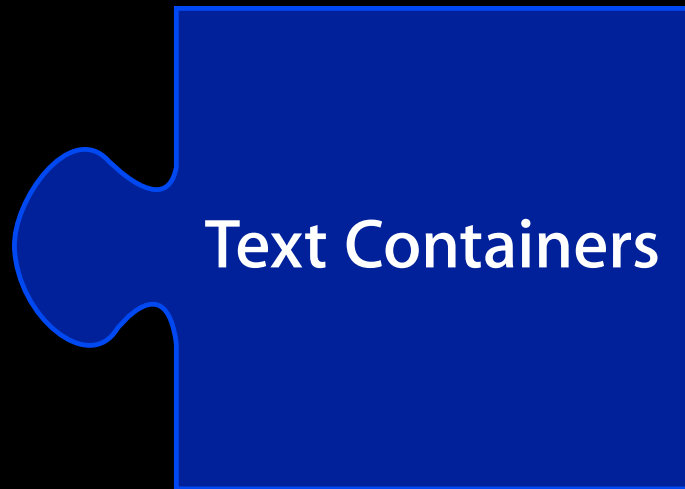
キーボードが入力特性を受け取る

Keyboard adopts traits



Keyboard adopts traits

入力が終わったときキーボードを隠すには
たとえば `[textField resignFirstResponder];`



UITextField

UITextView

Web Forms

ウェブページ上の
入力フォーム
(HTML5 ならカスタム化可能)

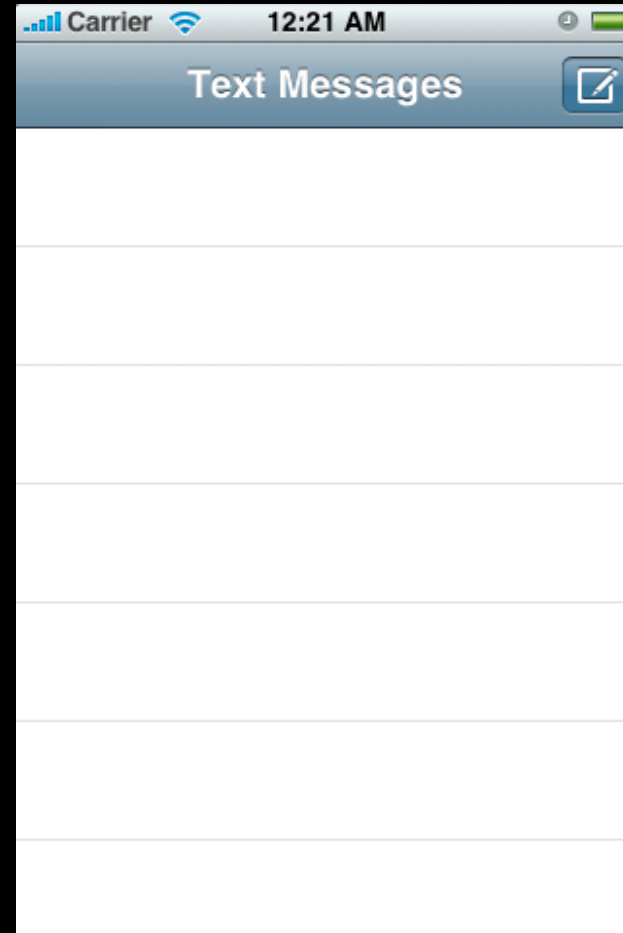
Demo: Text Input

Presenting Content Modally

一時的に（ユーザからの何らかの操作を期待して）
コンテンツを表示すること（いわゆるダイアログ）

Presenting Content Modally

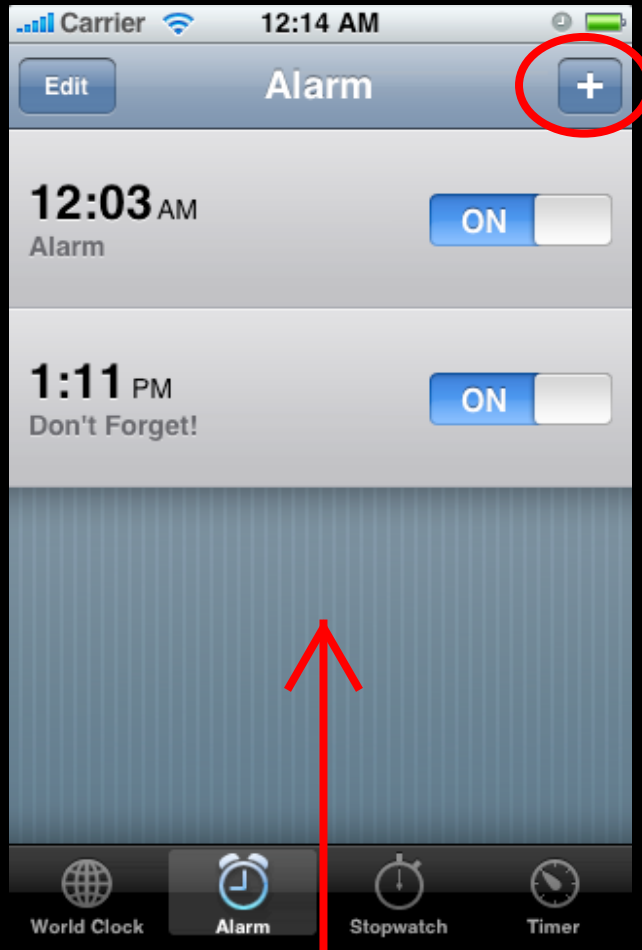
- For **adding** or **picking** data



Presenting Content Modally

ダイアログ的な
コンテンツ表示

- For adding or picking data データの追加あるいは選択



時刻設定ダイアログ



入れ子になったダイアログ

Modal View Controller によるダイアログ表示

Presenting a View Controller

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



ダイアログを開放する（閉じる）には...

Dismissing a View Controller

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



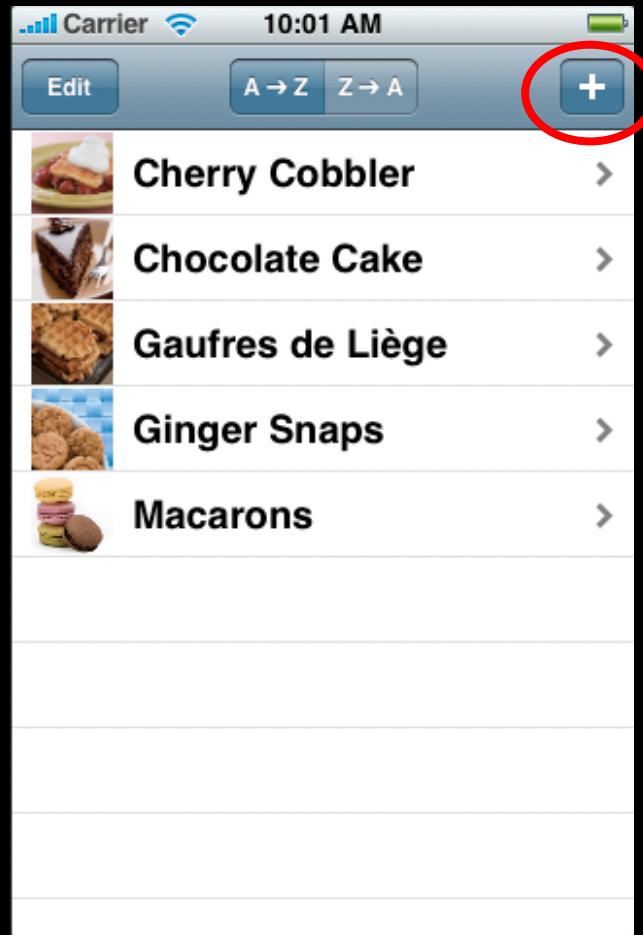
ダイアログを開放する（閉じる）には...

Dismissing a View Controller

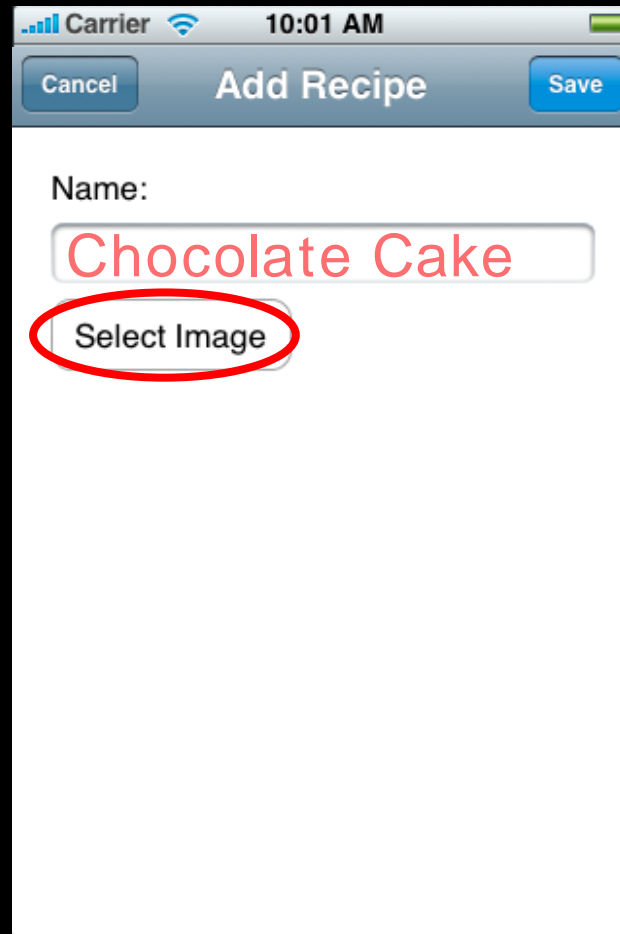
```
// Recipe list view controller  
- (void)didAddRecipe {  
    [self dismissModalViewControllerAnimated:YES];  
}
```



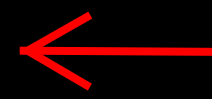
Separate Navigation Stacks



Separate Navigation Stacks



ダイアログの
スタックに
チョコケーキの
レシピ画面を
新規挿入
(この裏側に)



Separate Navigation Stacks



写真を選択したあとは
チョコケーキのレシピ
に戻るようになれる

ダイアログを閉じるには

Dismissing a Modal View Controller

- Who should do it? 表示したコントローラが閉じるのがよい
- 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

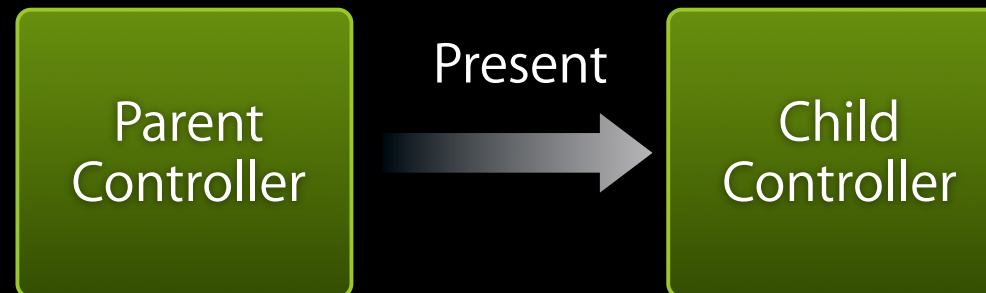
- Who should do it?
- Best practice is for the **same object** to call present and dismiss
- **Define delegate methods** for the presented controller
 - Tell the delegate when the presented controller is done
 - The delegate makes the call to dismiss



Parent
Controller

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



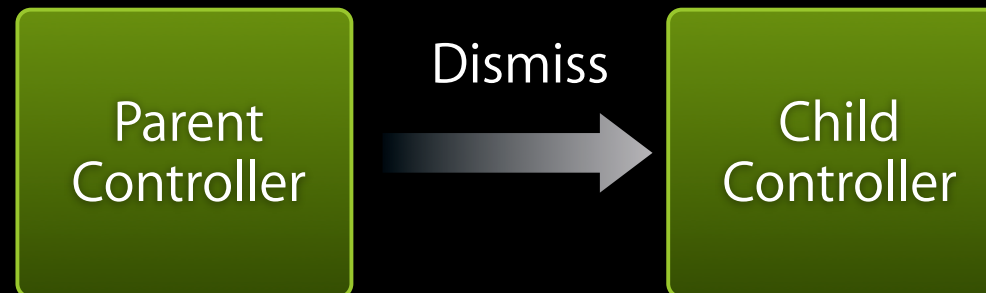
Dismissing a Modal View Controller

- 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



Parent
Controller

Demo: Presenting Content Modally

Questions?