

---

# iPad ヒューマンインターフェイスガイド ライン



2010-04-03



Apple Inc.  
© 2010 Apple Inc.  
All rights reserved.

本書の一部あるいは全部を Apple Inc. から書面による事前の許諾を得ることなく複製複製（コピー）することを禁じます。また、製品に付属のソフトウェアは同梱のソフトウェア使用許諾契約書に記載の条件のもとでお使いください。書類を個人で使用する場合に限り1台のコンピュータに保管すること、またその書類にアップルの著作権表示が含まれる限り、個人的な利用を目的に書類を複製することを認めます。

Apple ロゴは、米国その他の国で登録された Apple Inc. の商標です。

キーボードから入力可能な Apple ロゴについても、これを Apple Inc. からの書面による事前の許諾なしに商業的な目的で使用すると、連邦および州の商標法および不正競争防止法違反となる場合があります。

本書に記載されているテクノロジーに関しては、明示または黙示を問わず、使用を許諾しません。本書に記載されているテクノロジーに関するすべての知的財産権は、Apple Inc. が保有しています。本書は、Apple ブランドのコンピュータ用のアプリケーション開発に使用を限定します。

本書には正確な情報を記載するように努めました。ただし、誤植や制作上の誤記がないことを保証するものではありません。

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
U.S.A.

アップルジャパン株式会社  
〒163-1450 東京都新宿区西新宿  
3丁目20番2号  
東京オペラシティタワー  
<http://www.apple.com/jp/>

App Store is a service mark of Apple Inc.

Apple, the Apple logo, Finder, iPhone, iPod, iPod touch, iTunes, Mac, Mac OS, and Spotlight are trademarks of Apple Inc., registered in the United States and other countries.

iPad is a trademark of Apple Inc.

Apple Inc. は本書の内容を確認しておりますが、本書に関して、明示的であるか黙示的であるかを問わず、その品質、正確さ、市場性、または特定の目的に対する適合性に関して何らかの保証または表明を行うものではありません。その結果、本書は「現状有姿のまま」提供され、本書の品質または正確さに関連して発生するすべての損害は、購入者であるお客様が負うものとします。

いかなる場合も、Apple Inc. は、本書の内容に含まれる瑕疵または不正確さによって生じる直接的、間接的、特殊的、偶発的、または結果的損害に対する賠償請求には一切応じません。そのような損害の可能性があらかじめ指摘されている場合においても同様です。

上記の損害に対する保証および救済は、口頭や書面によるか、または明示的や黙示的であるかを問わず、唯一のものであり、その他一切の保証にかわるものです。Apple Inc. の販売店、代理店、または従業員には、この保証に関する規定に何らかの変更、拡張、または追加を加える権限は与えられていません。

一部の国や地域では、黙示あるいは偶発的または結果的損害に対する賠償の免責または制限が認められていないため、上記の制限や免責がお客様に適用されない場合があります。この保証はお客様に特定の法的権利を与え、地域によってはその他の権利がお客様に与えられる場合もあります。

# 目次

<b>序章</b>	<b>はじめに 7</b>
	この書類の構成 7
<b>第1章</b>	<b>iPadの主な機能と特徴 9</b>
	iPadデバイスの特徴 9
	新しいUI要素と動作 10
<b>第2章</b>	<b>iPhoneアプリケーションからiPadアプリケーションへ 11</b>
	iPhoneアプリケーションを変換するための設計方法 11
	ケーススタディ：iPhoneのメール(Mail)からiPadのメール(Mail)へ 12
<b>第3章</b>	<b>iPadユーザ体験ガイドライン 17</b>
	すべての向きをサポートすることを目標とする 17
	対話機能を強化する（単に機能を追加しない） 19
	情報階層をフラットにする 19
	フルスクリーントランジションを減らす 22
	共同作業とつながり 22
	質感とリアルさを高める 23
	魅力的なグラフィックスでユーザを楽しませる 24
	ユーザインターフェイスのコントロールを強調しない 26
	モダリティを最小限にする 26
	リストを見直す 26
	複数の指によるジェスチャを検討する 27
	一部のモーダルタスクにPopoverを使用することを検討する 27
	モーダルタスクでは複雑さを制限する 28
	ファイル処理操作を意識させない 29
	必要な場合にのみユーザに保存を求める 29
	Toolbarコンテンツを最上部へ移動する 30
	瞬時に起動する 31
	いつでも停止できる準備をしておく 32
	カスタムアイコンの作成 32
	美しいアプリケーションアイコンを作成する 33
	Spotlightの検索結果用アイコンを作成する 34
	「設定(Settings)」アイコンを作成する（「設定(Settings)」を提供する場合） 34
	カスタムドキュメントアイコンを提供する 35
	それぞれの方向用に起動画像を作成する 38
	確立されている原則に従う 38

**第4章 iPad UI要素ガイドライン 39**

---

- バー 39
  - ステータスバー 39
  - Navigation Bar 41
  - Tab Bar 42
  - Toolbar 43
- コンテンツビュー 44
  - Popover 44
  - Split View 47
  - Text View 48
- コントロール 49
  - 日付と時刻のピッカー、ピッカー 49
  - Infoボタン 50
  - ページインジケータ 50
  - Search Bar 50
  - Segmented Control 51
- Alert、Action Sheet、およびモーダルビュー 52
  - Action Sheet 52
  - Alert 54
  - モーダルビュー 55
- 編集メニューの追加 56
- キーボードのカスタマイズ 57

**改訂履歴 書類の改訂履歴 59**

---



## 第2章 iPhoneアプリケーションからiPadアプリケーションへ 11

---

- 図 2-1 iPhoneの「メール(Mail)」は簡素化された電子メール機能を一連の画面で表現する 13
- 図 2-2 iPadの「メール(Mail)」はどの方向でもうまく動作する（横長表示） 14
- 図 2-3 iPadの「メール(Mail)」はどの方向でもメッセージ内容にフォーカスする（縦長表示） 15
- 図 2-4 iPadの「メール(Mail)」は削除するようにマークしたメッセージを紙の束のように表示する 16

## 第3章 iPadユーザ体験ガイドライン 17

---

- 図 3-1 「設定(Settings)」は右側のペインでNavigation Barを使用する 20
- 図 3-2 「メール(Mail)」は左側のペインでNavigation Barを使用する 21
- 図 3-3 「メール(Mail)」はアカウント情報やメールボックスの情報を縦長モードで表示するためにPopoverを使用する 21
- 図 3-4 iTunesではSegmented Controlを使用してコンテンツについての複数の視点を提供する 22
- 図 3-5 iTunesはTab Barを使用してコンテンツのカテゴリを提供する 22
- 図 3-6 「連絡先(Contacts)」はユーザの期待どおりに動作するリアルなアドレスブックを表現する 23
- 図 3-7 リアルな外観、高品質な素材感は「メモ(Notes)」の効果を高める 25
- 図 3-8 iPhoneの「メール(Mail)」はToolbarで重要な機能を提供する 30
- 図 3-9 iPadの「メール(Mail)」は画面の最上部に機能を移動している 31
- 図 3-10 ホーム(Home)画面に表示される前の単純なアプリケーションアイコン 33
- 図 3-11 ホーム(Home)画面に表示された単純なアプリケーションアイコン 34
- 図 3-12 デフォルトのカスタムドキュメントアイコンはアプリケーションアイコンを表示する 35
- 図 3-13 64x64ピクセルのドキュメントアイコンの処理の前後 36
- 図 3-14 320x320ピクセルのドキュメントアイコンの処理の前後 37

## 第4章 iPad UI要素ガイドライン 39

---

- 図 4-1 ステータスバーはiPadの背景に溶け込んでいる 40
- 図 4-2 Navigation Barにはナビゲーションコントロールおよびその他のコントロールを含められる 41
- 図 4-3 Tab Barはサブタスク、ビュー、モードを切り替える 42
- 図 4-4 横向きの場合のTab Barに、縦向きの場合と同じ数のタブを表示する 43
- 図 4-5 Toolbarは画面やビューのオブジェクトに基づいて動作するコントロールを提供する 43
- 図 4-6 Popoverは一時的なビューで選択肢または機能を提示する 45
- 図 4-7 Split Viewには2つの関連するビューが含まれる 47



- 図 4-8 日付と時刻のピッカーによって複数の部分から成る値の選択が簡単になる 49
- 図 4-9 Search Barはユーザからの入力を受け取る 50
- 図 4-10 Scope Barはユーザが検索を絞り込むのに役立つ 51
- 図 4-11 Segmented Controlを使用して異なる視点やビューへのアクセスを提供できる 51
- 図 4-12 Segmented ControlはPopoverでは情報をフィルタリングしたりモードを切り替えたりするのに有効 52
- 図 4-13 Action SheetはPopover内に表示される 53
- 図 4-14 アニメーション化されたAction Sheetには一般的に「キャンセル(Cancel)」ボタンが付いている 54
- 図 4-15 モーダルビューでは自己完結するタスクを表示する 55

# はじめに

---

この文書『iPadヒューマンインターフェイスガイドライン』では、iPadとiPhone OS 3.2の機能を最大限に活用する優れたiPadアプリケーションを設計するために従うべきガイドラインについて説明します。

iPadアプリケーションに関しては、この文書のガイドラインが『iPhone Human Interface Guidelines』のガイドラインを補完し、場合によってはそちらに優先します。また、この文書を読んでも、『iPhone Human Interface Guidelines』を読んでいなければ、iPadを含むiPhone OSベースのすべてのデバイスで適用すべき情報が欠落することになります。

iPhone OSが初めてのデベロッパの方は、iPhone Reference Libraryを参照してプラットフォームに慣れるところから始めることを強くお勧めします。iPhone OSプログラミングが初めてのデベロッパの場合は、以下の文書を読むことから始めてください。

- *Your First iPhone Application*
- *iPhone Application Programming Guide*

iPhoneアプリケーションのユーザインターフェイスの設計が初めてのデベロッパであれば、以下の文書を読むことから始めてください。

- *iPhone Human Interface Guidelines*

これらの文書のほとんどのガイダンスは、iPad、iPhoneおよびiPod touchのすべてに等しく適用されます。それぞれの間で違いがある場合、『iPadヒューマンインターフェイスガイドライン』と『iPad Programming Guide』がデバイス固有のガイダンスを提供します。

## この書類の構成

『iPadヒューマンインターフェイスガイドライン』は、次の各章で構成されています。

- **“iPadの主な機能と特徴”** (9 ページ) では、iPhoneと比較して異なる特徴と機能に焦点をあててデバイスの概要を説明します。
- **“iPhoneアプリケーションからiPadアプリケーションへ”** (11 ページ) では、iPhoneアプリケーションをiPad用に変換するいくつかの方法について説明します。
- **“iPadユーザ体験ガイドライン”** (17 ページ) では、iPadアプリケーションで素晴らしいユーザ体験を提供するのに役立つガイドラインを示します。
- **“iPad UI要素ガイドライン”** (39 ページ) では、iPadアプリケーションでのUI要素の使い方について説明します。

## 序章

はじめに

# iPadの主な機能と特徴

iPadはiPhone OS 3.2を実行します。そしてiPadアプリケーションはiPhoneアプリケーションで使用したものと同一UIKitの多くのビューとコントロールを使用します。このため、デベロッパのiPhoneアプリケーション開発における経験（および『*iPhone Human Interface Guidelines*』の知識）が大変役立ちます。

iPadは、この馴染みのある基盤の上に築かれていますが、iPhoneのユーザ体験とは大きく異なる新たな体験をユーザにもたらしめます。大きな画面と魅力的で高度な対話型インターフェイスを通じて、iPadは新たな分類のアプリケーションを作るための今までにない機会をデベロッパに提供します。

デベロッパは、iPadユーザ体験を理解し、その見識に基づいてデバイスに合ったアプリケーションを設計することに開発時間の多くを割くことになります。

## iPadデバイスの特徴

iPadには、アプリケーションのユーザインターフェイスに大きな影響を与える次のような新しい特徴が取り入れられています。

- 1024x768ピクセルの大きな画面サイズ
- デフォルトのデバイスの向き、またはユーザの期待に従ったデバイスの向きがない
- 外付けキーボードを接続してオンスクリーンキーボードの代わりにそれを使うオプションをユーザに提供
- ユーザがデバイスをDockに接続できる機能

開発の方向性を見出しやすいように、以下にiPadとiPhoneに共通の特徴を示します。

- メモリは限られている
- 一度に実行するアプリケーションは1つ
- 環境設定は「設定(Settings)」アプリケーションで設定できる
- デバイスの向きを変えられる
- オンスクリーンユーザヘルプは最小限で控えめである
- アプリケーションはマウスクリックではなく手を使ったジェスチャに反応する
- デバイス上ではネイティブなソフトウェア、Web専用のソフトウェア、およびそれらのハイブリッドソフトウェアが実行される
- アートワークのビット深度が決まっている。具体的には24ビット（赤、緑、青についてはそれぞれ8ビット）に8ビットのアルファチャンネルを加えたものである。一般的にはPNG形式が推奨される

## 新しいUI要素と動作

iPhone OS 3.2ではUIKitに新しいUI要素が含まれています。また、以前のバージョンでも利用できた要素のいくつかに新しい動作が定義されています。

- **Split View**。このiPad専用の要素は、たとえばデータをマスタ・詳細の形式やソースリストの形式で表示するなど、一度に1つ以上のビューを画面上に表示するために使用できます。Split Viewは情報階層のフラット化に役立つため、iPadアプリケーションで情報を整理するためによく使う要素です。アプリケーションでSplit Viewを使用する方法の詳細については、“[iPadユーザ体験ガイドライン](#)”（17 ページ）を参照してください。使用方法のガイドラインについては、“[Split View](#)”（47 ページ）を参照してください。
- **Popover View**。このiPad専用のビューは、メインビューの内容に関係する追加の情報、コントロール、または選択肢を一時的に表示するために使用します。Popover Viewの主な利点は、メインインターフェイスに常に表示する必要のない情報や選択肢を含めることができる点です。Popover Viewの使用法の詳細については、“[一部のモーダルタスクにPopoverを使用することを検討する](#)”（27 ページ）を参照してください。使用方法のガイドラインについては、“[Popover](#)”（44 ページ）を参照してください。
- **結果リストボタン**。システムに用意されているこのボタンは、Search Barからの検索の結果を表示するために使用できます（使用方法のガイドラインについては、“[Search Bar](#)”（50 ページ）を参照してください）。
- **モーダルビューに新しい表示スタイルが追加された**。フルスクリーン、部分スクリーン、フォームなどのスタイルを使用して、アプリケーションのユーザ体験や視覚的なデザインにこれまで以上に合ったモーダルインターフェイスを表示できます（使用方法のガイドラインについては、“[モーダルビュー](#)”（55 ページ）を参照してください）。
- **Toolbarを表示できる場所が増えた**。Toolbarは、画面の上部または下部に配置できます。また、Split ViewまたはPopoverの中でToolbarを使用することもできます（使用方法のガイドラインについては、“[Toolbar](#)”（43 ページ）を参照してください）。
- **編集メニューにカスタム項目を表示できる**。標準の「カット(Cut)」、「コピー(Copy)」、「ペースト(Paste)」、「選択(Select)」、および「全選択(Select All)」の各コマンドを補完したり置き換えたりするメニュー項目を提供できます（使用方法のガイドラインについては、“[編集メニューの追加](#)”（56 ページ）を参照してください）。
- **キーボードビューをカスタマイズできる**。システムに用意されているキーボードビューを、カスタムボタンを含むカスタムビューで置き換えることができます（使用方法のガイドラインについては、“[キーボードのカスタマイズ](#)”（57 ページ）を参照してください）。
- **キーボードビューには、カスタム入力アクセサリを含めることができる**。補助キーボードコントロールを別のビューとして提供することで、ユーザにそれをタップさせ、アプリケーション固有のコンテンツを入力させることができます。
- **カスタムテキストビューは、複数の形式のテキストをサポートし、高度な編集機能を提供できる**。ワープロ機能を提供し、文字入力にはスペルチェックやオートコンプリート機能をサポートできます。

# iPhoneアプリケーションからiPadアプリケーションへ

すでに作成したiPhoneアプリケーションがある場合、iPadを最大限利用できるようにアプリケーションを修正する方法を知る必要があります。iPhone OSに精通していれば比較的簡単にコードを変更するのに役立ちますが、ユーザインターフェイスやユーザ体験を発展させることは難しいかもしれません。

**重要：** iPhoneアプリケーションは、修正しなくてもそのままiPad上では互換モードで実行します。これにより、ユーザはiPad上でデベロッパのiPhoneアプリケーションにアクセスできますが、ユーザが望むデバイス固有の体験は提供されません。

この章では、iPhoneアプリケーションのユーザインターフェイスを修正する一般的な方法を説明します。従う必要がある具体的なガイドラインについては、“iPadユーザ体験ガイドライン”（17 ページ）を参照してください。iPhoneアプリケーションのコードをiPad用に変換するのに役立つ情報については、『iPad Programming Guide』を参照してください。

## iPhoneアプリケーションを変換するための設計方法

iPhoneアプリケーションのユーザインターフェイスとユーザ体験を修正し、iPadの大きな画面を利用してユーザが期待する強化された対話性を提供することは非常に重要です。これを行う具体的な方法は、iPhoneアプリケーションの仕様に応じて異なります。

**ゲームや没入型のiPhoneアプリケーション**は、データ駆動型ではなく体験駆動型であるため情報アーキテクチャに大きな変更は必要ありません。ただし、一般的にゲームは、iPad上で魅力的な体験を提供するためにはアートワークとインタラクションに大きな改訂が必要になります。ゲームを改訂する場合は次の項目を考慮してください。

- アートワークを可能な限り高品質にすることが非常に重要です。iPadでは、ユーザは深く探求できる贅沢なビジュアル環境をゲームが提供してくれるものと期待します。アートワークを単純に拡大するのではなく、解像度を高め、ユーザがアプリケーションに没頭できるようにあらゆる機会をとらえてアートワークを精細にしていきます。
- 同様に、視覚効果の品質も高めます。クリアできわめて精密な、実際の事象やアクションにそっくりな描写はユーザの情緒反応を高め、ゲーム固有の環境に入り込む感覚を強めます。
- 基本的なユーザ体験が変わらないようにしつつ、ユーザとアプリケーションの対話を増やす方法を追求するべきです。たとえば次のようなことです。
  - マルチプレイヤー機能を追加することを検討する。
  - カメラアングル、つまみ、スイッチ、ダイヤルを追加することで目にしたり動かしたりするものを増やしたり、ゲームの駒やゲームプレイをカスタマイズする方法を増やしたりする。
  - 理にかなっていれば、複数の指によるジェスチャをサポートする。特にカスタムコントロールと連携させることができる場合にそうします。

**iPhone生産性型アプリケーション**は、表現力豊かなUIや強化されたユーザ体験に加えて情報階層の再設計を必要とする場合があります。生産性型アプリケーションを改変する場合は次の項目を念頭に置いてください。

- 1つの画面で、ユーザがより多くの情報を目にし、より多くの作業をこなせるようにすることを目指します。ユーザが目的を達成するために多数の画面をたどる必要がないようにします（これを実現するいくつかの方法については、“[情報階層をフラットにする](#)”（19ページ）を参照してください）。
- ユーザインターフェイスは、目の作業からユーザの気が散らないように、できるだけ控えめにすべきです。しかしだからといって、ユーザインターフェイスをきれいにすることを避けるべきではありません。生産性型アプリケーションが重要な作業を行うために使用するものとしても、ユーザはiPad上での体験を楽しむことを期待しています。作業になじむようにわずかにカスタマイズした要素を使用したり、ユーザが操作するオブジェクトをよりリアルに見えるようにしたりすることを検討します。また、データを文字ではなく視覚的に表現するなど、より魅力的な方法で表示することも検討します。
- iPadアプリケーションの対話性を強化する様々な方法を探します。主たる作業に常に焦点をあてておくのが望ましいですが、多くの場合、ユーザが見てコントロールできるデータを増やすことは良い考えです。

**ユーティリティ型アプリケーション**は、より大きな画面で利用できるように想定を見直す必要があります。一般に、画面全体をフリップすることはお勧めしないため、多くのユーティリティ型アプリケーションはその対話モデルを変更する必要があります。ユーティリティ型のiPhoneアプリケーションの場合は以下を検討します。

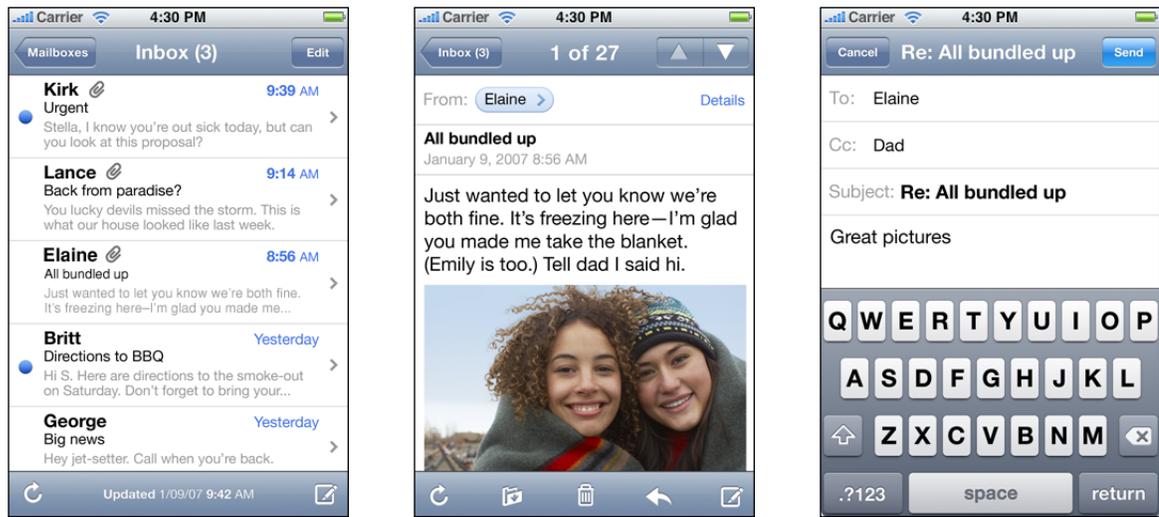
- iPhoneの複数の異なる画面のコンテンツを組み合わせて、iPad上では1つの画面にする方法を検討します。
- 目の作業について、より詳細な情報をユーザに提供することを考えます。iPhoneユーティリティ型アプリケーションでコンテンツのサマリビューを提供している場合は、ユーザが同じ画面上で展開ビューを表示できるようにすることも考えられます。
- ユーザがコンテンツと対話する方法を増やします。たとえば、ユーザがリアルタイムにグラフを操作したり、データシミュレーションを表示したり、または表示されているコンテンツの詳細をカスタマイズしたりできるようにすることは理にかなっています。

iPhoneアプリケーションのスタイルにかかわらず、“[iPadユーザ体験ガイドライン](#)”（17ページ）に説明されているiPad固有のガイドラインには必ず従ってください。

## ケーススタディ：iPhoneのメール(Mail)からiPadのメール(Mail)へ

「メール(Mail)」はiPhoneの最初の組み込みアプリケーションの1つです。ユーザは、大量の情報を整理する、分かりやすく簡素化された方法や、その使いやすさ、そしてそのスケーラビリティを高く評価しています。

図 2-1 iPhoneの「メール(Mail)」は簡素化された電子メール機能を一連の画面で表現する



iPadの「メール(Mail)」は、iPhoneの場合と同様の中核機能を、次のような改良されたユーザ体験を通して提供します。

- 広くなった画面上のスペースにユーザのメッセージを表示
- リアルさを出すための見映えの改善
- いつでもアクセスできてかつ邪魔にならない強力な構成ツールと編集ツール
- 最低限の内容の変更でユーザに必要なものを一か所で提供する、単一の、整頓された視覚的に安定した画面

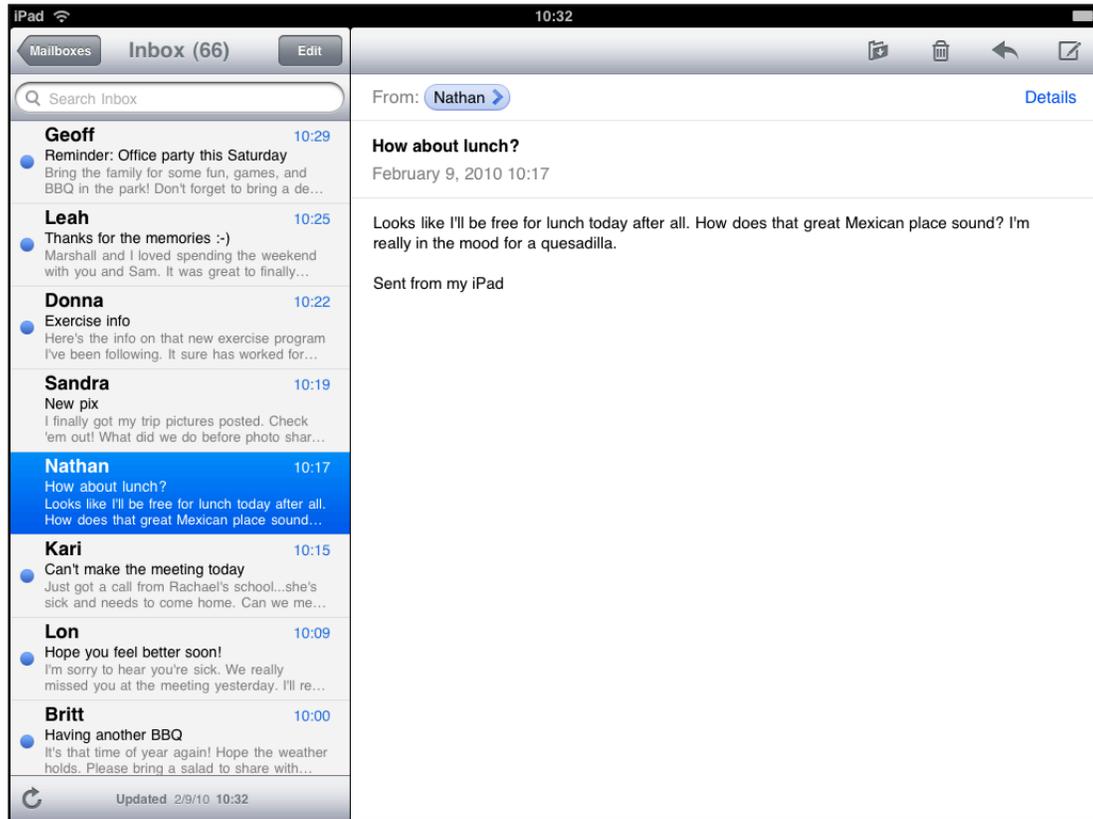
iPhoneの「メール(Mail)」とiPadの「メール(Mail)」の違いは、それぞれのデバイスのユーザ体験の相違を反映しています。iPhoneの「メール(Mail)」は、モバイルユーザが列に並んでいる間や会議に向かう間に電子メールを操作しやすいように設計されています。iPadの「メール(Mail)」は移動中に使用するユーザにも十分効果的ですが、より深い使い方をユーザに提供します。

iPadの「メール(Mail)」はデバイスに合わせてユーザ体験が調整されていますが、ユーザが慣れ親しんだ中核機能に変わりはないことに注目してください。また、個別の機能の場所や効果も根拠なく変更されていません。ツールバー項目やメールボックスの構造は、iPhoneとiPadのどちらでも実質的には同じであるため、iPhoneの「メール(Mail)」ユーザはiPadの「メール(Mail)」でも簡単にそれらを認識し、すぐにその使い方を理解します。

電子メールのモバイル体験を向上するために、iPadの「メール(Mail)」はiPhoneの「メール(Mail)」のUIを主に次の2つの点で進化させています。

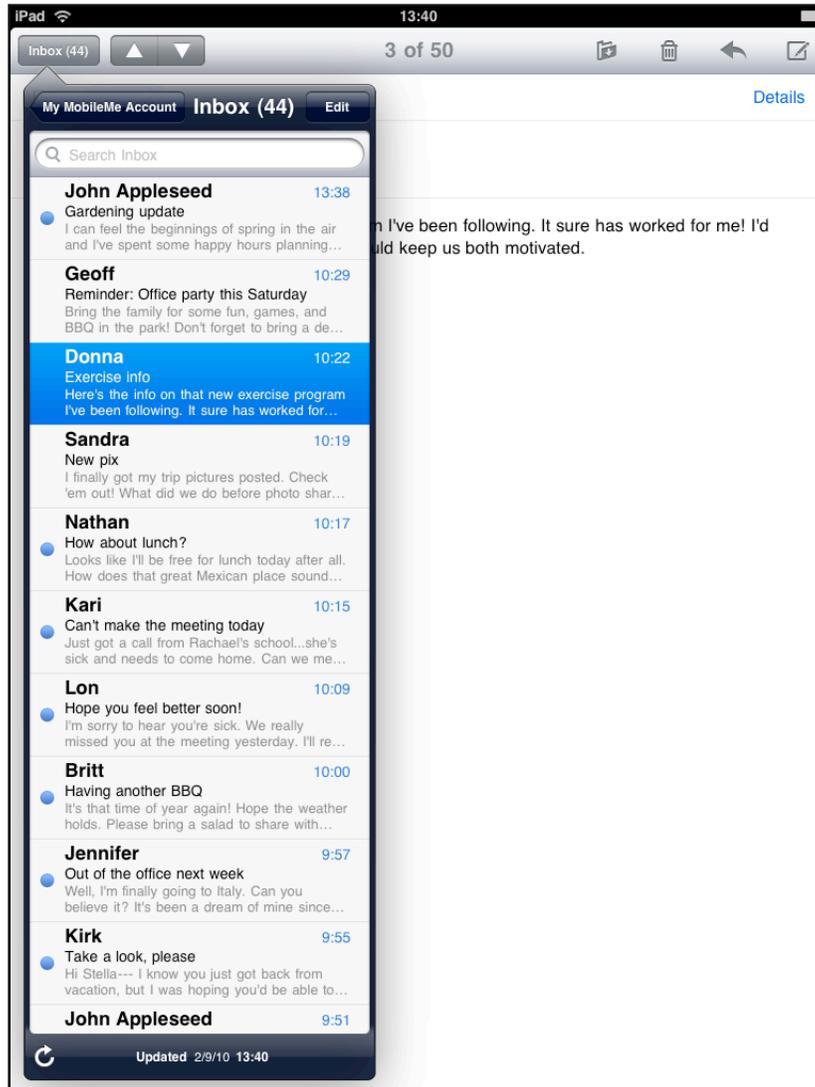
**デバイスの向きに対する拡張サポート。**ユーザは、iPadの「メール(Mail)」を4方向すべてで使用できます。横長レイアウトは縦長レイアウトに比べて少々異なりますが、ユーザはどの方向でも関心のあるすべての機能に簡単にアクセスできます。図2-2には横長モードの「メール(Mail)」を、図2-3には縦長モードの「メール(Mail)」を示しています。

図 2-2 iPadの「メール(Mail)」はどの方向でもうまく動作する (横長表示)



**メッセージ内容へのフォーカスを強化。** iPadの「メール(Mail)」は、画面の大部分をカレントメッセージのために確保しています。これには、メッセージビューの上にToolbarを移動して、メッセージ内容に使用できる縦方向のスペースを増やすことも含まれます。余分にできたスペースを使うことで、ユーザはこれまでよりも少ないスクロール回数で長いメッセージを読めます。また、メッセージリストを表示している場合でもユーザはカレントメッセージの多くの部分を表示したままにできます。

図 2-3 iPadの「メール(Mail)」はどの方向でもメッセージ内容にフォーカスする（縦長表示）

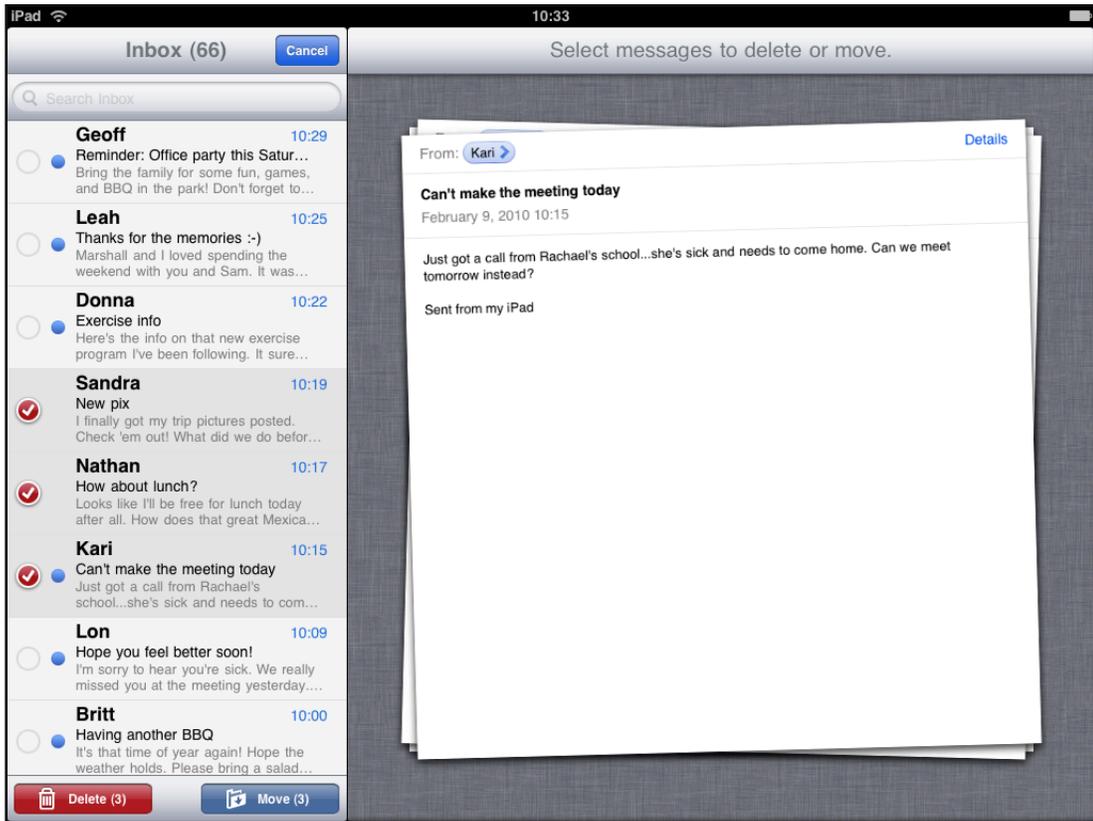


**フラットな階層。** iPadの「メール(Mail)」は、メッセージの上位にあるすべてのレベルを別のUI要素に結びつけて、「アカウント(Account)」>「メールボックス(Mailbox)」>「メッセージリスト(Message List)」>「メッセージ(Message)」という階層を実質的にフラット化しています。横長モードではこの要素はSplit Viewの左側のペイン（図 2-2に示すとおり）になり、縦長モードではこの要素はPopover（図 2-3に示すとおり）になります。

**フルスクリーンランジションの大幅減。** ほとんどの階層は画面上の個別の要素で利用できるため、ユーザは必要な項目のほとんどに1つの画面でアクセスできます。ユーザが階層内をドリルダウンするとき、ランジションが起こるのは画面全体ではなく、Split View内またはPopover内です。

**リアルなメッセージ。** ユーザが削除するメッセージをマークすると、メッセージは物理的な紙のようにメッセージビューの上に滑り込んできます。ユーザが削除するメッセージを追加して選択すると、メッセージは端が不均等に少しずれた状態のリアルな紙の束として表示されます（図 2-4に示します）。

図 2-4 iPadの「メール(Mail)」は削除するようにマークしたメッセージを紙の束のように表示する



# iPadユーザ体験ガイドライン

コンテンツと対話機能は、iPadユーザ体験の中で最も重要です。最も優れたiPadアプリケーションは、次の3点を実現することによってコンテンツおよび対話機能を高めています。

- ユーザが求めるコンテンツに集中できるように、アプリケーションUIを意識させない。
- コンテンツを美しく、そしてしばしばリアルに見えるように表現する。
- コンテンツとの高度な対話を可能にするため、デバイスの機能を最大限に利用する。

コンテンツと対話機能の重要性が、この章に示すガイドラインを特徴付けます。ここで説明する内容を念頭において、iPadアプリケーションを設計するようにしてください。

## すべての向きをサポートすることを目標とする

すべての向きで実行できることは、iPadアプリケーションを成功させる重要な要素です。大きな画面は、デバイスを横向きに回転させてもっと広い範囲を表示させたいというユーザの欲求を緩和します。また、ユーザはほとんど目に付かないデバイスの枠やホーム(Home)ボタンの位置にあまり注意を払わないため、デバイスがデフォルトの向きを持つとは考えません。こうしたことからユーザは、現在使用しているデバイスの向きでアプリケーションが問題なく実行されることを期待します。アプリケーションはできるかぎりすべての向きで優れた体験を提供することで、どのような向きからもiPadと対話できることをユーザに示すべきです。

横長モードと縦長モードのサイズの違いは、画面内にUIを収める方法に大きく影響します。回転に反応する具体的な方法はさまざまありますが、次のガイドラインに従うように努めてください。

**主たるコンテンツに焦点が保たれるようにする。**これは最優先事項です。ユーザがアプリケーションを使うのは興味のあるコンテンツを表示したりコンテンツとの対話を行ったりするためです。向きを変えたときにコンテンツへの焦点が変わると、ユーザはアプリケーションに対する制御を失ったと感じます。

**補足の情報または機能の表示方法を変更することを検討する。**最も重要なコンテンツに常に焦点が確実に当たっているようにする必要がありますが、回転に反応するときに、二次的なコンテンツの提示方法を変更することができます。

たとえば「メール(Mail)」では、アカウントリストとメールボックスのリストが二次的コンテンツを構成します（主たるコンテンツは選択されたメッセージです）。横長モードでは二次的コンテンツがSplit Viewの左側のペインに表示され、縦長モードではPopoverに表示されます。あるいは、横長モードで矩形のゲームボードが表示されるゲームを考えてみます。縦長モードでは、ゲームはボードを再描画して画面に合わせる必要があり、その結果、ボードの上下に余分な空間が表示される場合があります。スペースに合わせてゲームボードを垂直に伸ばしたり、空いている空間をそのまま残したりする代わりに、ゲームはその余った空間に補足情報やオブジェクトを表示することができます。これらの例はどちらも主たるコンテンツにユーザの焦点を保ち、アプリケーションの主要な機能を変更することなく現在の画面サイズを最大限に活用します。

**過度または不要なレイアウトの変更は避ける。**大きなiPad画面では、すべての向きで同じUIのレイアウトを提供することが可能です。たとえば横長モードの時に画像をグリッドに並べて表示する場合、縦長モードの時に同じ情報をリストにして表示する必要はありません（グリッドのサイズは調整が必要かもしれません）。二次情報のレイアウトを変更することがあったとしても、すべての向きで一貫性のある体験を提供することに注力してください。すべての向きで同じ体験を提供することで、デバイスを回転した場合でもユーザは使用パターンを保つことができます。

**可能であれば、回転時に情報の書式を変えたりテキストの折り返しを変更したりしないようにする。**すべての向きで同じ書式を保つようにします。特に、ユーザがテキストを読んでいる場合、デバイスを回転した時に読んでいた場所を見失わないようにすることが重要です。

やむなく書式を変える場合は、アニメーションを使ってユーザがその変更を追跡しやすくします。たとえば、向きを変えたときにテキストカラムを追加したり削除したりする必要がある場合には、カラムの移動を見えないようにして、新しい配置がフェードインする方法を選ぶと良いでしょう。回転時の動作を適切な設計にするには、実物を現実の世界で物理的に取り扱ったとしたらコンテンツにどのような動作を期待するかを考えると良いでしょう。

**コンテンツを回転させるUI要素を提供したり、回転ジェスチャを定義したりしないこと。**代わりに、ユーザがデバイスを回転することによってコンテンツを回転できるようにします。

**それぞれの向きごとに起動画像を提供すること。**それぞれの向きごとに起動画像があれば、ユーザは現在のデバイスの向きにかかわらずスムーズなアプリケーションの起動を体験します。iPhoneのホーム(Home)画面とは違い、iPadのホーム(Home)画面はすべての向きをサポートしているので、ユーザは直前にアプリケーションを終了したときと同じ向きでアプリケーションを開始する可能性が高いでしょう。iPadの起動画像の詳細については、“[それぞれの方向用に起動画像を作成する](#)”（38ページ）を参照してください。

**アプリケーションをすべての向きで実行できるようにしないと決める前に、もう一度よく考えてみる**こと。ユーザは現在持っているiPadの向きでアプリケーションを使用することを期待しており、その期待に応えることが最善です。ただし、場合によっては、アプリケーションを縦長モードあるいは横長モードでのみ実行する必要があります。ある一定の向きでのみアプリケーションを実行することが不可欠な場合は、次のようにします。

- **デバイスの現在の向きにかかわらず、サポートする向きで起動すること。**たとえば、ゲームまたはメディア表示アプリケーションを横長モードのみで実行する場合、デバイスが現在縦長の向きであったとしても横長モードで起動するのが適切です。そうすることで、縦長の向きの状態でアプリケーションを起動した場合でも、ユーザはコンテンツを見るためにデバイスを横長の向きに回転する必要があることがわかります。
- **ユーザにデバイスを回転するように指示するUI要素の表示は避けること。**サポートする向きで起動すれば、UIを不必要に煩雑にすることなく、必要があればデバイスを回転する必要があることがユーザにははっきりと伝わります。
- **1つの向きの2つの可能性（左右または上下）のどちらもサポートすること。**たとえば、アプリケーションを横長モードでのみ実行する場合、ユーザはデバイスのホーム(Home)ボタンが右にある向きで持っていたとしても左にある向きで持っていたとしてもアプリケーションを使える必要があります。また、ユーザがアプリケーションの使用中にデバイスを180度回転させたとき、コンテンツを180度回転させて対応できるのが最善です。

**アプリケーションで、デバイスの向きの変化をユーザ入力と解釈する場合には、アプリケーション固有の方法で回転を処理できます。**たとえば、アプリケーションが、ユーザがデバイスを回転させることでゲームの駒を移動するゲームの場合、デバイスの回転に対して画面の回転で応えることはできません。このような場合には、必要な向きのどちらかで起動して、ユーザがアプリケーション

の主要タスクを開始するまでは、そのどちらにも切り替えられるようにする必要があります。その後、ユーザが主要タスクを開始したらすぐに、アプリケーション固有の方法でデバイスの動きに応じることができます。

## 対話機能を強化する（単に機能を追加しない）

最良のiPadアプリケーションは、明確に定義された限定されたタスクをユーザが実行する間、コンテンツと対話するための革新的な方法をユーザに提供します。主要タスクに直接関係しない機能で大きな画面をいっぱいにするという誘惑には耐えてください。iPadの画面が大きいからといって、iPhoneアプリケーションで削った機能をすべて復活するように誘っているわけではありません。

優れたiPadアプリケーションを作成するには、関係のない機能で主要タスクを弱体化させることなく、ユーザ体験を強化する方法に集中します。たとえば次のようにします。

- ユーザが読書したり読書リストを記録したりできる書籍閲覧アプリケーションは、大きな画面によってよりいっそう楽しめる読書体験を提供できます。読書リストを管理するために別の画面にユーザを移動させるのではなく、Popoverでリストを表示してそこにお気に入りの一節をコピーできるようにできます。また、ユーザがテキストへブックマークや注釈を加えられるようにしたり、ほかのユーザとリストを交換したりリストを一元管理するリポジトリに照らして進み具合を把握できるようにしたりできます。
- パイロットとして戦闘機を操るゲームであれば、メインビューに半透明のヘッドアップディスプレイをかぶせても良いでしょう。プレイヤーは、コックピットのリアルな操作盤をタップして、敵と交戦したりマップオーバーレイ上で自身の場所を表示したりできます。
- サッカーゲームでは、広くてリアルさを増したフィールドと、より詳細なキャラクターを表示して、ユーザがチームの管理やキャラクターのカスタマイズを行えるようにできます。ユーザがフィールドビューを表示したままでキャラクター情報を表示したりもできます。最後に、2人のユーザがマルチプレーヤモードを使用して、お互いのチームを戦わせるようにもできます。
- 脚本執筆アプリケーションでは、メインコンテキストから離れることなくプロットビューとキャラクタービューを切り替える方法を提供しても良いでしょう。ライターはこれらのビューを切り替えて、メインビューで執筆しながら細部を確認できます。

## 情報階層をフラットにする

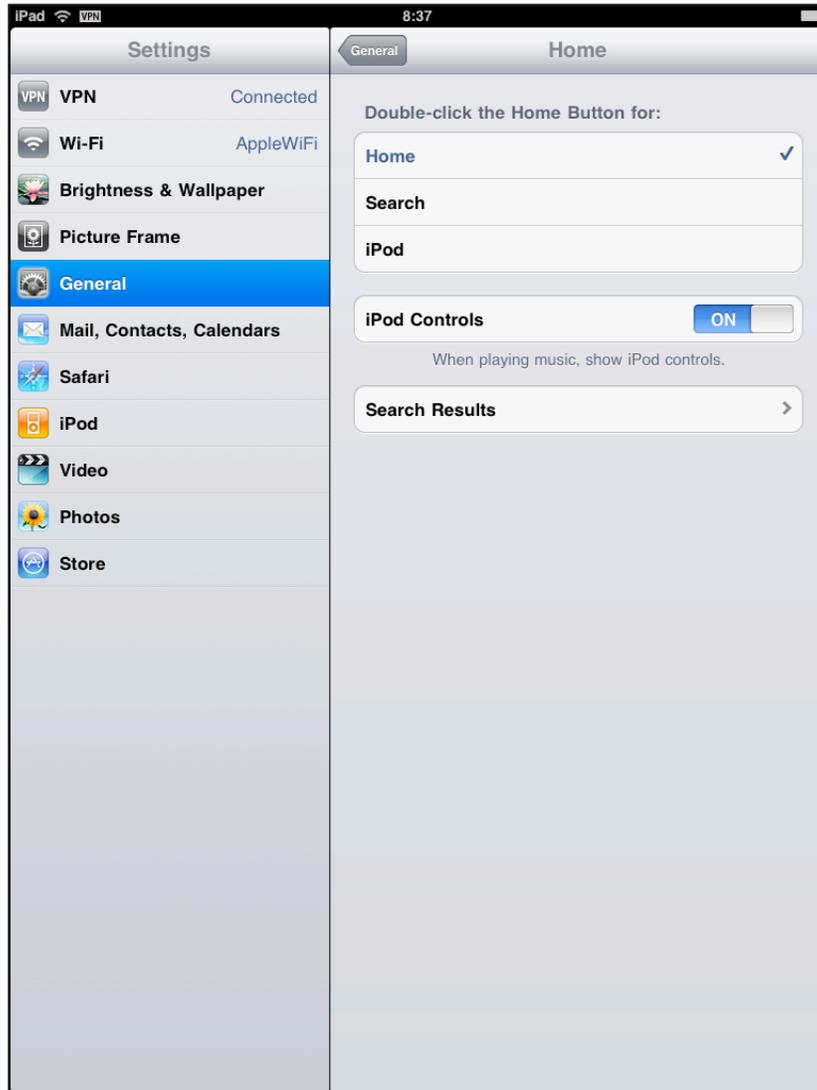
大きなiPad画面と新しいUI要素を使用して、ユーザが1か所でより多くの情報にアクセスできるようにします。1つの画面に情報を詰め込み過ぎるのは避けるべきですが、ユーザが必要なものを見つけるためにたくさんの異なる画面を閲覧しなければならないとユーザが感じないようにする必要があります。

一般に、メイン画面では主たるコンテンツに焦点を当て、追加情報や追加のツールはPopoverなどの補助ビューの中で提供します。これにより、ユーザは主たる作業のコンテキストから離れることなく、必要な機能に簡単にアクセスできるようになります。

大きなiPad画面と、Split ViewやPopoverなどのUI要素を使用すると、多くのiPhoneアプリケーションの1画面あたり1レベルの構造に代わる構造が得られます。たとえば、次のようにします。

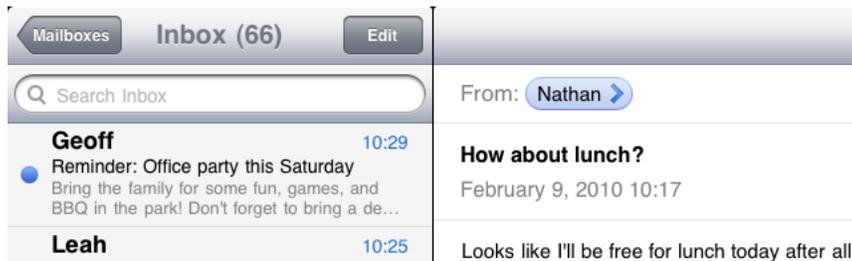
**Split Viewの右側のペインでNavigation Barを使用すると**、ユーザは、左側のペインに持続的に表示される最上位のカテゴリをドリルダウンできるようになります。これにより、2つのレベルが常に画面上に同時に表示されるため、少なくとも1レベル分だけ情報階層がフラット化されます。図 3-1 に示すとおり、「設定(Settings)」はこの方法でデバイスとアプリケーションの設定を表示します（使用方法のガイドラインについては、「[Split View](#)」（47 ページ）を参照してください）。

図 3-1 「設定(Settings)」は右側のペインでNavigation Barを使用する



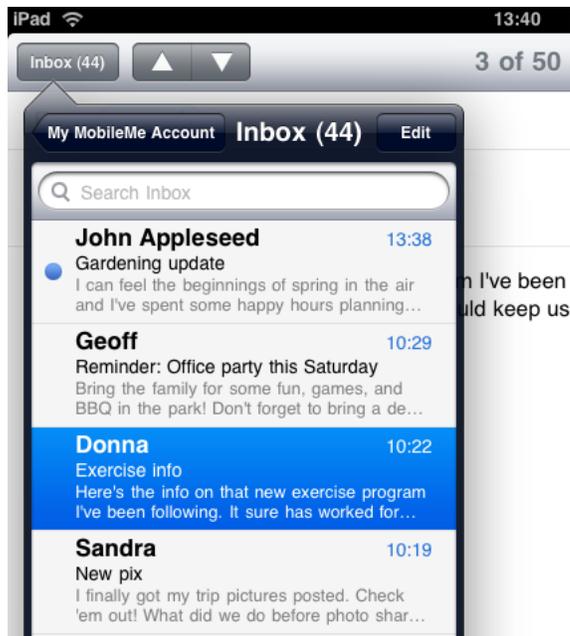
**Split Viewの左側のペインでNavigation Barを使用すると**、ユーザは、比較的浅い階層をドリルダウンできるようになります。そして、最も具体的な情報（階層内のリーフノード）を右側のペインに表示します。これもまた、一度に2つのレベルを表示することで階層がフラット化します。横長モードでの「メール(Mail)」は、この設計を使用してユーザのメールボックス階層を左側のペインに表示し（図 3-2に示します）、個別のメッセージを右側のペインに表示します（図 2-2（14 ページ）に示します）（使用方法のガイドラインについては、「[Navigation Bar](#)」（41 ページ）を参照してください）。

図 3-2 「メール(Mail)」は左側のペインでNavigation Barを使用する



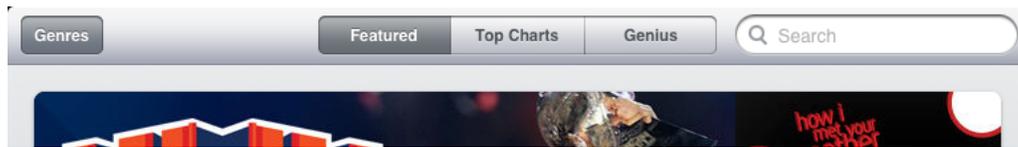
ポップオーバーを使用すると、画面上のオブジェクトに作用するアクションを有効にしたりツールを提供したりできます。Popoverは、現在の画面の一番手前にこれらのアクションとツールを一時的に表示できます。つまり、ユーザはアクションやツールを使用するために別の画面に移動する必要はありません。縦長モードでの「メール(Mail)」はユーザのアカウントやメールボックスの階層を表示するためにPopoverを使用します（図 3-3に示します）（使用方法のガイドラインについては、「[Popover](#)」（44 ページ）を参照してください）。

図 3-3 「メール(Mail)」はアカウント情報やメールボックスの情報を縦長モードで表示するためにPopoverを使用する



ツールバーでSegmented Controlを使用すると、コンテンツを異なる視点、あるいは異なる情報カテゴリで表示できます。この方法により、画面の一番上（または一番下）の1つのバーから、これらの視点またはカテゴリへのアクセスを提供できます。iTunesでは上端のツールバーでSegmented Controlを使用して、カテゴリ内のコンテンツについて各種の視点を提供しています（図 3-4に示します）（使用方法のガイドラインについては、「[Toolbar](#)」（43 ページ）および「[Segmented Control](#)」（51 ページ）を参照してください）。

図 3-4 iTunesではSegmented Controlを使用してコンテンツについての複数の視点を提供する



**Tab Barを使用すると**、異なる情報カテゴリや、異なるアプリケーションモード（頻度は低い）を表示できます。iPadアプリケーションでは、Tab Barはモード切替用ではなく、フィルタ切替用またはカテゴリ切替用に使用することが多いでしょう。図3-5に示すように、iTunesはTab Barを使用してメディアの異なるカテゴリにユーザがアクセスできるようにします。Tab Barを使用せずにさまざまな画面を切り替えられるようになるなら、情報アーキテクチャを変更して複数の、並行するモードを避けるだけの価値はあります（使用方法のガイドラインについては、“[Tab Bar](#)”（42 ページ）を参照してください）。

図 3-5 iTunesはTab Barを使用してコンテンツのカテゴリを提供する



## フルスクリーントランジションを減らす

視覚的なトランジションは、変化するコンテンツと密接に結び付けるようにします。埋め込まれている何らかの情報に変化があったときに、画面全体を新しい画面に切り替えるのではなく、ユーザインターフェイスの必要な領域のみ更新するようにします。一般的な規則として、画面ではなく個々のビューやオブジェクトのトランジションを行うようにします。ほとんどの場合、画面全体をフリップすることはお勧めしません。

フルスクリーントランジションの実行が少なければ少ないほど、ユーザは作業のどの段階にいるかを把握しやすいため、アプリケーションは視覚的に安定します。Split ViewやPopoverなどのUI要素を使用すれば、フルスクリーントランジションの必要性を減らせます。

## 共同作業とつながり

ユーザはiPadを個人向けのデバイスと見なしていますが、その都合の良いサイズは、ほかのユーザとの物理的な共同作業や共有も促します。

ユーザが、ほかのユーザと共同でアプリケーションを使う方法を考えてみましょう。1つのデバイスを物理的に共有すること、そして仮想的なデータを共有することの両方を含めるように考えを広げてください。たとえば、2人のユーザが画面上のゲーム盤で対面してゲームをプレイできると良いかもしれません。あるいは、バンドアプリケーションであれば、さまざまなユーザが1つのデバイスで別々の楽器を一緒にプレイできるようにしても良いでしょう。

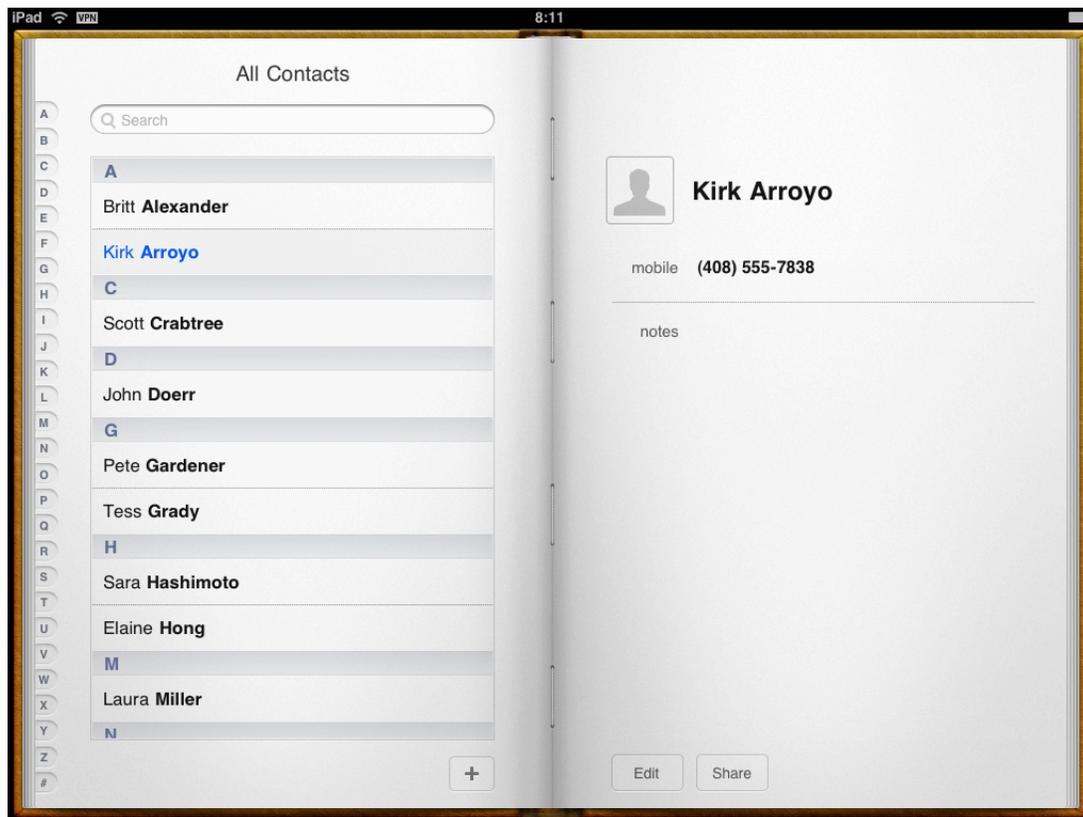
ユーザは、彼らにとって重要な情報をほかのユーザと共有できることを期待します。アプリケーションにおいてそれが理にかなっていれば、ほかのユーザとのやり取りや、意見、場所、ハイスコアなどの共有をユーザが簡単に行えるようにします。

ほとんどのアプリケーションは、アプリケーションを超えて、ユーザが使用するその他のツールとデータを共有できるようにすることで、価値を高めることができます。たとえば、iPadアプリケーションは、コンピュータのアプリケーションに対してモバイル機能を補完する役割を果たせます。あるいは、iPadアプリケーションは、iPadアプリケーションのユーザがiPhone版アプリケーションのユーザとやりとりできるようにもすることもできます。

## 質感とリアルさを高める

アプリケーションのリアルさ、物質的な質感はできる限り高めるようにします。アプリケーションの外観と動作が実物そっくりであればあるほど、ユーザはアプリケーションがどのように機能するかを理解しやすくなり、より楽しく使うことができます。たとえば、「連絡先(Contacts)」が表示するリアルなアドレスブック (図 3-6に示します) の使い方はすぐにわかります。

図 3-6 「連絡先(Contacts)」はユーザの期待どおりに動作するリアルなアドレスブックを表現する



アプリケーションにリアルな質感を加えることに取り組む際に、徹底した精度を得るための努力が必要なわけではありません。多くの場合、誇張したり強調したりした何らかの描写のほうが、忠実に表現したものよりもリアルで、より多くの意味を伝えると感じられます。オブジェクトやシーンを設計する際には、それが、ユーザとの対話とアプリケーションの本質を表現する機会であると考えてください。

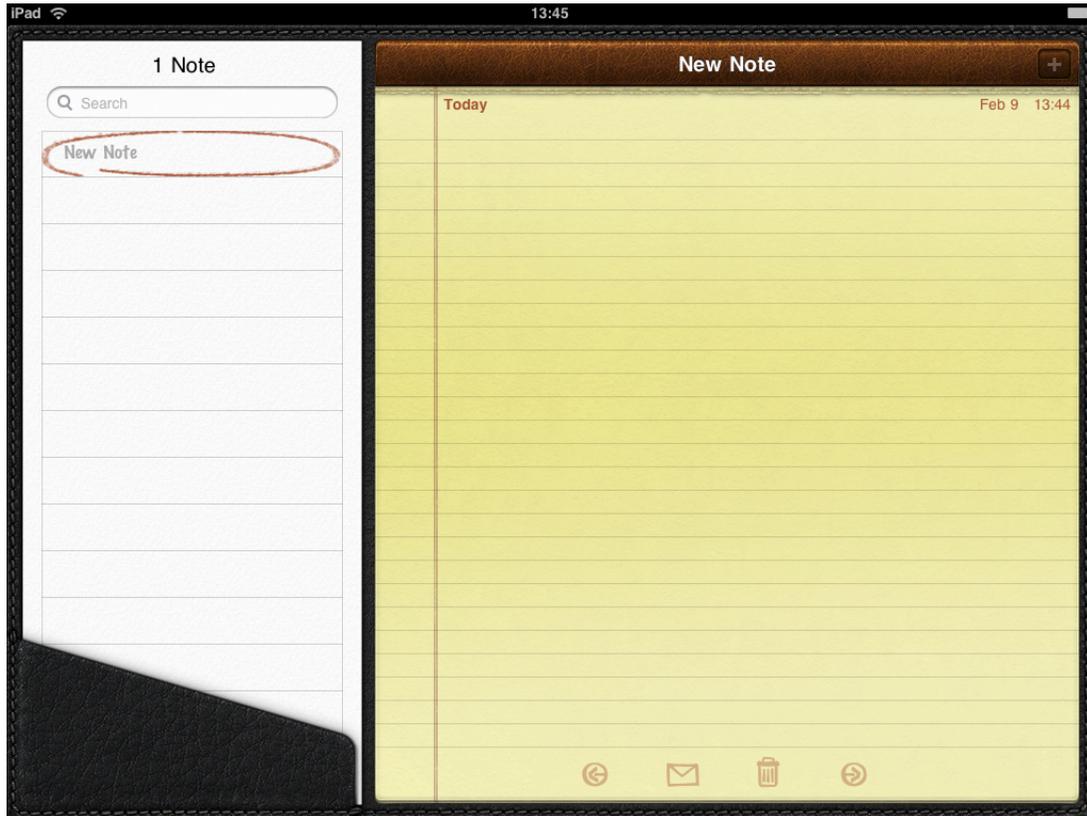
アプリケーションのリアルさをさらに強化するには、アニメーションを使用します。一般に、外見の精度よりも動作の精度を高める努力をすることのほうが重要です。なぜなら、ユーザは外観の創作的演出は許容しますが、物理的な法則を無視するような動きを見た場合は混乱を感じるためです。仮想的なビューとコントロールが、それらの表現する物理的なオブジェクトとコントロールの動作をできるだけ再現するようにしてください。分かりやすいアニメーションは、アプリケーションに具体的で物理的なリアルさを与え、それに時間を費やしたいというユーザの印象を強めます。

## 魅力的なグラフィックスでユーザを楽しませる

iPadの高解像度の画面は、ユーザをアプリケーションに引き付け、最もシンプルな作業に価値を持たせる、豊かで美しい、魅力的なグラフィックをサポートします。iPadはアプリケーションのネットワークを提示するので、ユーザが賞賛するような素晴らしいグラフィックを作成するプロのアーティストを雇うことを検討すべきです。

アプリケーションの知覚価値を高める1つの方法は、高品質な外観や貴重な素材を再現することです。たとえば、木、革、金属の効果が適したアプリケーションであれば、素材の見た目がリアルで価値のあるものにすることに時間を費やします。図 3-7に示すように、「メモ(Notes)」では、きめ細かい革と実物のような縫い目を持つ外観が再現されています。

図 3-7 リアルな外観、高品質な素材感は「メモ(Notes)」の効果をも高める



まだ高解像度なアートワークではない場合、アートワークを再作成する必要があるかもしれませんが。ほとんどの場合、アートワークを拡大することは長期的な解決策としてはお勧めしません。代わりに、縮小する前に深みと細部を加えられるように、必要とされるものよりも大きなサイズでアートワークを作成するようにします。これは、オリジナルのアートファイルの大きさが必要なサイズの倍数である場合、特に役立ちます。そして、画像編集アプリケーションで適切なグリッドサイズを使用すれば、縮小したアートファイルの鮮明さを保ち、必要なレタッチとシャープの回数を減らせます。

すべてのテキスト、エフェクト、画像の更新に加え、画面のサイズを示すハードコードの値はすべて必ずコードから取り除きます。

必要であれば、既存の起動画像を更新して追加の画像を作成します（[“それぞれの方向用に起動画像を作成する”](#)（38 ページ）を参照）。

大きなアプリケーションアイコンを作成します（[“美しいアプリケーションアイコンを作成する”](#)（33 ページ）を参照してください）。

## ユーザインターフェイスのコントロールを強調しない

ユーザが関心を持っている情報に対する控えめな枠となるようにアプリケーションのUIをデザインすることで、ユーザがコンテンツに集中できるようにします。アプリケーションコントロールの数や目立ち方を最低限にすることで、それを意識させないようにします。「写真(Photos)」は、半透明なバーにいくつかの控えめなコントロールを配置することで、これを行っています。

アプリケーションのグラフィカルなスタイルに巧みに一体化するカスタムコントロールを作成することを検討します。こうすることで、コントロールを目立たせることなく見つけやすくなります。

また、ユーザが操作をやめてしばらく時間が経ったコントロールをフェードし、ユーザが画面をタップしたときに再表示することを検討します。また、アプリケーションのその他のUIをフェードさせても良いでしょう。こうすると、ユーザが目にしたコンテンツに、より多くの画面スペースを与えることができます。たとえば、「写真(Photos)」は、ユーザが操作をやめてからしばらくするとコントロールとバーがフェードし、ユーザがコンテンツに集中しやすくなります。

## モーダリティを最小限にする

可能であれば、ユーザが何らかの作業を行ったり応答をしたりするためにモーダル環境にない回数で済ませるべきです。iPadアプリケーションでは、ユーザが非線形にアプリケーションを操作できるようにすべきです。モーダリティは、ユーザの作業の流れに割り込み、特定の作業経路の選択を強いることで、この自由を妨げます。

モーダリティが適しているのは以下のような場合です。

- ユーザの注意を引くことが重要な場合。
- ユーザのデータを曖昧な状態に残さないように、タスクを完了させる（あるいは明示的に破棄する）必要がある場合。

## リストを見直す

リスト（つまり、Table View）は、iPhoneアプリケーションで大量の情報を効率良く表示するための一般的な方法です。リストはiPadアプリケーションでも非常に有用ですが、デベロッパはこの機会に、同じ情報をさらに豊かな表現で提示できないか検討すべきです。たとえば次のようにします。

- アプリケーションをさらに実物に近づけられないか検討する。たとえば、iPhoneでは「連絡先(Contacts)」は簡素化されたリストですが、iPadでは「連絡先(Contacts)」は美しく実物のような外観と質感を持つアドレスブックです（これについては、[図 3-6](#)（23 ページ）を参照してください）。
- 一部の情報をリスト項目ではなくオブジェクトとして表示することを検討する。たとえば、iPod アプリケーションは、アルバムのカバーサムネイルのグリッドにアルバムを表示します。また「メール(Mail)」では、ユーザが削除するものとしてマークしたメッセージは実際の紙の束のように表示されます（これについては、[図 2-4](#)（16 ページ）を参照してください）。
- ほかのビューに埋め込むことによってリストの幅を制限する。たとえば、画面の幅全体を使うのではなく、Popoverの中やビューの背面にリストを表示するのが適切な場合があります。

- 可能な場合には、iPhoneアプリケーションで行ったのとまったく同じ形式でリスト情報を表示しないようにする。iPadではリストの幅がずっと広いいため、iPhoneのリストの行を埋めるコンテンツはiPadではまばらに見えてしまいます。広くなったスペースを利用して、それぞれの行に追加の情報や詳細を提供すると良いでしょう。

## 複数の指によるジェスチャを検討する

iPadの大きな画面では、複数のユーザによるジェスチャを含め、複数の指によるカスタムジェスチャに対する自由度が大きくなっています。複雑なジェスチャは、どのアプリケーションにも適しているわけではありませんが、ゲームやコンテンツ制作の環境など、ユーザが長時間使用するアプリケーションの体験を豊かなものにすることができます。標準でないジェスチャは知るのが難しく、そうしたジェスチャを採用するとしても、それがアクションを実行する唯一の方法とすべきではないことは常に念頭においておきましょう。

採用するジェスチャが、アプリケーションの機能とユーザの期待という観点から見て必ず理にかなっているようにします。たとえば、ユーザが頻繁に実行してすぐにでも完了することを期待するような重要なタスクを可能にするアプリケーションの場合、標準のジェスチャだけを採用すべきでしょう。しかし、アプリケーションが、特定の使い方を要求するリアルなコントロールを備えていたり、ユーザが探索するような環境を提供したりする場合、カスタムジェスチャや複数の指によるジェスチャが適している場合があります（標準のジェスチャの詳細については、『*iPhone Human Interface Guidelines*』の「Support Gestures Appropriately」を参照してください）。

## 一部のモーダルタスクにPopoverを使用することを検討する

通常、Popoverやモーダルビューが開いている場合はユーザはメインビューとやりとりできないという意味では、Popoverとモーダルビューは似ています。しかし、モーダルビューは常にモーダルであるのに対し、Popoverは以下のように2つの異なる方法で使用できます。

- **モーダルモード**。この場合、Popoverによって周りの画面領域が暗くなり、明示的にPopoverを閉じることが求められます。この動作はモーダルビューと非常によく似ていますが、Popoverの外観のほうが軽い印象を与える傾向があります。
- **非モーダルモード**。この場合、Popoverによって周りの画面領域は暗くならず、ユーザがPopoverの外側をタップするとPopoverが閉じます。この動作によって、非モーダルモードのPopoverは、独立の状態ではなくアプリケーションの別のビューであるように見えます。

さらに、Popoverにはそれを開くためにユーザがタップした元のコントロールや領域を指す矢印が常にあります。この視覚的な関連性は、ユーザが直前のコンテキストを思い出すのに役立ちます。これはまた、モーダルなPopoverを、表示元となった場所を示すことなく画面を支配するモーダルビューよりも一時的な状態であるかのように感じさせます。

iPhoneアプリケーションでモーダルビューを使用して自己完結型のタスクを可能にしている場合、代わりにPopoverを使用すると良いでしょう。Popoverをどのようなときに使うべきか判断するには、以下の問いを考えてみてください。

- タスクは異なるタイプの入力が必要とするか。必要とする場合には、Popoverを使用します。  
Popoverまたはモーダルビューのどちらの場合でもキーボードは利用できますが、ピッカーまたは選択肢のリストを表示するにはPopoverが適しています。

- タスクでは、ビュー階層をユーザがドリルダウンする必要があるか。必要とする場合には、Popoverを使用します。  
ユーザがメインビューと混同する可能性を低減できるため、Popoverのフレームは複数ページの表示に適しています。
- ユーザがタスクを終了する前にメインビューで何か行う可能性があるか。そうであれば、非モーダルモードでPopoverを使用します。  
ユーザが非モーダルモードのPopoverの裏にあるメインビューを見ることができ、メインビューの中をタップすることで非モーダルモードのPopoverを閉じることができるため、ユーザがPopoverのタスクを中断して、また戻ることができるようにする必要があります。
- 対象タスクがかなり込み入っており、アプリケーションの主要機能の1つを反映するものか。そうであれば、モーダルビューを使用すると良いでしょう。  
モーダルビューによって生じるコンテキストの大きな切り替えにより、ユーザはタスクを終了するまでそのタスクに集中できます。モーダルビュースタイルのほとんどで提供される大きな画面スペースは、ユーザが多くの入力を簡単に行えます。  
逆に、タスクがアプリケーション機能の重要な部分を反映するものの込み入ったものでなければ、モーダルモードのPopoverを選択するほうが良い場合もあります。これは、Popoverが視覚的に軽く、頻繁に実行するようなタスクではそのほうが快適だからです。
- タスクはセットアップ作業のように一度だけの実行か、またはタスクの実行自体非常にまれなものか。そうであれば、モーダルビューの使用を検討します。  
ユーザは、タスクの実行が一度だけであったり非常にまれな場合、現在のコンテキストに留まることにそれほどこだわりません。

補助ツールの提供など、Popoverにはほかにもいくつかの使用方法があります（使用方法の完全なガイドラインについては、“[Popover](#)”（44ページ）を参照してください）。また、iPadアプリケーションではPopoverの内部にAction Sheetを表示することにも注意してください（詳細については、“[Action Sheet](#)”（52ページ）を参照してください）。

モーダルビューを使用する場合、使用できるさまざまな表示スタイルについても必ず目を通してください（詳細については、“[モーダルビュー](#)”（55ページ）を参照してください）。iPadアプリケーションでは、有効にする必要があるモーダルタスクに最も適した表示スタイルを選択できます。

## モーダルタスクでは複雑さを制限する

モーダルビューは、コンテキストの切り替えが明確で一時的であるため、ユーザは自己完結型サブタスクをモーダルビューで実行することを歓迎します。しかしサブタスクが複雑すぎると、モーダルビューに移ったときに中断した主要タスクをユーザが見失ってしまう可能性があります。このリスクは、モーダルビューがフルスクリーンの場合や、モーダルビューに複数の従属ビューまたは状態が含まれる場合に大きくなります。

モーダルタスクは、短時間で済むようにし、対象となる範囲を狭く保つようにします。ユーザにモーダルビューをアプリケーション内のミニアプリケーションとして経験させてはいけません。モーダルビューの階層を伴うモーダルタスクの作成には特に注意してください。ユーザは元の位置に戻るまでの道筋を見失ったり忘れたりするからです。モーダルタスクにサブビューを含めなければならぬ場合、単独の明確な経路を通して階層をたどれるようにして、循環が生じるのを避けます。

モーダルタスクを終了するための明確で安全な方法を必ず提供するようにします。ユーザがモーダルビューまたはPopoverを閉じるとき、その作業の結果は常に予測できるべきです。

タスクで複数のモーダルビューを必要とする場合は、最上位レベルの配下にあるビューの「完了(Done)」ボタンをタップすると何が起こるかをユーザが確実に理解できるようにしておきましょう。タスクを検討して、サブビューの「完了(Done)」ボタンでタスクのサブビューの部分だけを終了するのか、すべてのタスクを終了するのかを決定します。可能であれば、混乱を招く可能性があるため、サブビューに「完了(Done)」ボタンは追加しないようにします。

## ファイル処理操作を意識させない

iPadアプリケーションではユーザにファイルを作成させたり操作させたりできますが、iPad上のファイルシステムをユーザが理解する必要があるというわけではありません。

iPadにはMacOSXのFinderのようなアプリケーションはなく、コンピュータで行うようなファイルの操作をユーザに要求すべきではありません。特に、以下に示すような、ファイルのタイプや場所について考えさせるような状況にユーザを直面させるべきではありません。

- ファイルの階層を表示する「開く(Open)」ダイアログまたは「保存(Save)」ダイアログ
- ファイルのパーミッションの状態についての情報

代わりに、ドキュメントを扱うiPadアプリケーションでは、ユーザがコンテンツをアプリケーション内のオブジェクトとして捉えるように促すべきです。

ユーザがドキュメントを作成したり編集したりできるようにiPadアプリケーションの場合、何がしかのドキュメントピッカーを用意してユーザが既存のドキュメントを開いたり新しいドキュメントを作成したりできるようにするのが適切です。理想的には、たとえば次のようなドキュメントピッカーです。

- **高度にグラフィカルなドキュメントピッカー。**ユーザは、画面上にあるドキュメントの視覚的な表現を見て、必要なドキュメントを簡単に識別できる必要があります。
- **可能な限り最小限のジェスチャでユーザが望むことを行えるようにするドキュメントピッカー。**たとえば、ユーザが既存のドキュメント群を回転ラックのように水平にスクロールして、目的のドキュメントを1タップで開けるようにします。
- **新規ドキュメント作成の機能を持ったドキュメントピッカー。**新規ドキュメント作成のためにユーザをどこか別の場所へ移動させるのではなく、ドキュメントピッカーでユーザがプレースホルダー画像をタップすると新規ドキュメントが作成されるようにします。

## 必要な場合にのみユーザに保存を求める

ユーザは、明示的にキャンセルしたり削除したりしない限り、自分の作業は必ず保存されると確信できなければなりません。ユーザのドキュメント作成やドキュメント編集を支援するアプリケーションの場合、明示的な保存のアクションをユーザが行わなくても良いようにします。iPadアプリケーションでは、ユーザによる入力を責任を持って定期的に保存することと、別のドキュメントのオープン時またはアプリケーション終了時に保存することのどちらも行わなければなりません。

アプリケーションの主要機能がコンテンツ作成ではないものの、ユーザが情報の表示と情報の編集の間を切り替えられるようにする場合、変更を保存するかどうかユーザに確認することは理にかなっています。このようなシナリオでは、多くの場合、情報を表示するビューに「編集(Edit)」ボタンを用意すると都合が良いです。ユーザが「編集(Edit)」ボタンをタップしたときに、そのボタンを「保存(Save)」ボタンに変更して「キャンセル(Cancel)」ボタンを追加できます。「編集(Edit)」ボタンを変更することで、自分が編集モードにいることをユーザが再認識しやすくなります。また「キャンセル(Cancel)」ボタンは、加えた変更を保存せずに編集を終了する機会をユーザに与えます。

ユーザは意図せずPopoverを閉じる可能性があるため、一般に（ユーザが作業をキャンセルするのでない限り）Popoverでユーザが入力した情報は保存するようにします。Popoverの使用方法に関する具体的なガイドラインの詳細については、「[Popover](#)」（44 ページ）を参照してください。

## Toolbarコンテンツを最上部へ移動する

iPhoneアプリケーションにToolbarがある場合、Toolbarを最下部のまま残すのではなく画面の最上部に移動することを検討します。iPadでは画面の幅が広がったので、Toolbarのすべての機能を最上部の1つのToolbarで提供できるでしょう。これにより、フォーカスしているコンテンツに与えられる縦方向のスペースが増えます。

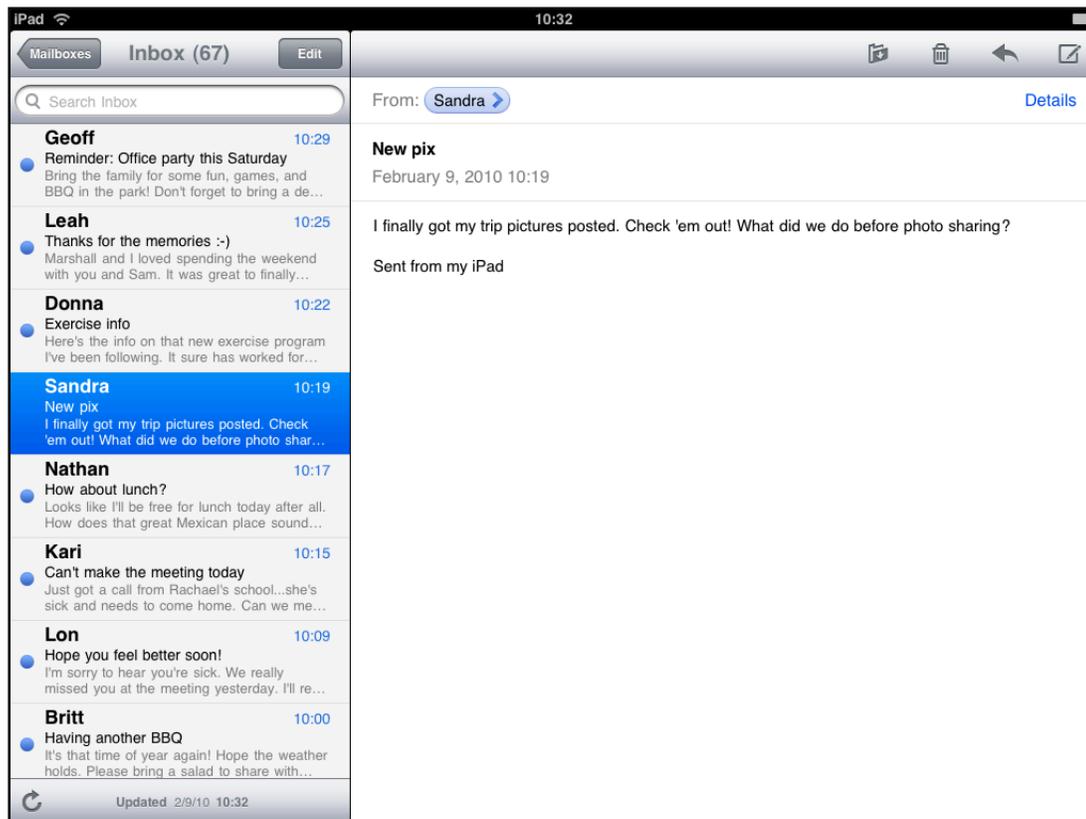
たとえば、iPhoneの「メール(Mail)」では、メッセージを表示している間、Toolbarを使って「更新(Refresh)」、「移動(Organize)」、「ゴミ箱(Trash)」、「返信(Reply)」および「作成(Compose)」などのアクションへのアクセスをユーザに提供しています（図 3-8を参照してください）。

図 3-8 iPhoneの「メール(Mail)」はToolbarで重要な機能を提供する



iPadの「メール(Mail)」では、図 3-9に示すように、これらのアクションの1つを除きすべてへのアクセスをメッセージの上のToolbarで提供します。「更新(Refresh)」コントロールは、縦長モードではPopover、横長モードではSplit Viewの左ペインにあるメールボックスのリスト内にあります。

図 3-9 iPadの「メール(Mail)」は画面の最上部に機能を移動している



## 瞬時に起動する

iPadアプリケーションは、ユーザがアプリケーションを遅滞なく使い始められるように、可能な限り瞬時に起動するべきです。起動の際にiPadアプリケーションは、次を実行すべきです。

- **現在の向きでのアプリケーションの最初の画面によく似た起動画像を表示する。**これにより、アプリケーション起動時間が感覚的に短縮され、デバイスが動作していることがわかるのでユーザを安心させられます（詳細については、“[それぞれの方向用に起動画像を作成する](#)”（38ページ）を参照してください）。
- **アプリケーションの起動を遅くする「・・・について(About)」ウインドウやスプラッシュ画面の表示を避ける。**ブランド情報を表示することは禁じられていませんが、ユーザは毎回アプリケーションを開始するたびにその内容を目にするということを覚えておく必要があります。スプラッシュ画面を表示する必要があると考える場合は、その画面があまり長く表示され続けず、ユーザが何も操作しなくても消えるようにしてください。最も重視すべきなのは、アプリケーションの起動を素早く快適にすることです。
- **アプリケーションを最後に実行したときの状態を復元する。**アプリケーション内の最後にいた場所に到達するまでのステップを、ユーザが覚えておく必要がないようにするべきです。
- **ユーザに設定情報の入力を求めるのは避ける。**代わりに、以下を行います。

- **80%のユーザのニーズに応えるよう重点的に取り組む。** そうすれば、アプリケーションはすでにほとんどのユーザの期待どおりに動作するようにセットアップされているため、大半のユーザは設定を行う必要がありません。一部のユーザだけが必要とするような機能や、ほとんどのユーザが一度しか必要としないような機能がある場合は除外します。
- **システムからできる限り多くの情報を得る。** 組み込まれているアプリケーションやデバイス設定においてユーザが指定する情報を使用できる場合は、それらの値をシステムに照会します。ユーザにそれらの値の再入力を要求してはいけません。
- **ユーザに入力を求める前に、アプリケーションを使うことによってユーザが恩恵を得られるようにする。** ユーザがアプリケーションの全機能を使えるようになる前にユーザから情報を得なければならない場合は、最初に、ユーザの入力を必要としない事柄をユーザが実現できるようにします。その後、必要に応じて情報を求め、できるだけ早くその情報を格納して再度その情報を求める必要がないようにします。

## いつでも停止できる準備をしておく

iPhoneアプリケーションと同様に、iPadアプリケーションは、ユーザが別のアプリケーションを開くためにホーム(Home)ボタンを押すと停止します。したがって、快適な停止体験を提供するために、iPadアプリケーションは以下を実行するべきです。

- **できるだけ早く、かつ妥当な頻度でユーザデータを保存する。** これは、終了通知がいつ届くかかもしれないためです。頻繁に保存すれば、アプリケーションはより素早く終了し、ユーザは「保存(Save)」ボタンをタップする必要がありません（**“必要な場合にのみユーザに保存を求め”**（29 ページ）を参照してください）。
- **停止時に現在の状態をできるだけ詳細なレベルまで保存する。** アプリケーションを再開したとき、ユーザは以前のコンテキストに戻ることを期待しています。たとえば、Split Viewを使用する場合、マスタペインの現在の選択を記憶しておき、ユーザがアプリケーションを再開したときにそれを表示します。

## カスタムアイコンの作成

どのアプリケーションも、ほかに独自のアートワークを持たない場合でも、独自のアプリケーションアイコンを提供する必要があります。また、Spotlightの検索結果に表示する小さなアイコンも提供する必要があります。アプリケーションによっては、独自のバーアイコンや、設定アイコン、独自のドキュメントタイプを表すアイコンを提供する必要がある場合があります。アプリケーションのほかのアートワークとは異なり、これらすべてのカスタムアイコンは、iPhone OSで適切に表示できるように特定の条件を満たす必要があります。

**重要：** コード内でこれらのアイコンを正しく指定する方法については、『*iPhone Application Programming Guide*』の「Application Icons」を参照してください。

**注：** Navigation Bar、Toolbar、およびTab Barでは、可能であればシステムに用意されているボタンおよびアイコンを使用すべきです。これらの場所で使用するカスタムアイコンを設計する必要がある場合は、『*iPhone Human Interface Guidelines*』の「Icons for Navigation Bars, Toolbars, and Tab Bars」を参照してください。

## 美しいアプリケーションアイコンを作成する

App Storeでは、アプリケーションを潜在顧客に紹介するのは、アプリケーションアイコンです。アプリケーションアイコンは、ユーザのホーム(Home)画面でユーザがアプリケーションを使うたびに必ず最初と最後に目にするものです。人を引き付け、アプリケーションを表現する魅力的なアイコンを作るために時間を費やすことには相応の価値があります。

**72x72ピクセルのアプリケーションアイコンを作成する。** このサイズであれば、アプリケーションの内容を表現するのに申し分ない広さがあります。アプリケーションアイコンの作成に関する1つの考え方は、アプリケーションのポスターをデザインしていると想像してみることです。アプリケーションの特徴や目的を紹介し、対話方法やストーリー展開を示唆し、アプリケーションの雰囲気を出す場面をデザインします。このようなアイデアを使って、ユーザがホーム(Home)画面で見て楽しめる表現豊かで精密なアイコンを作成します。

iPhoneアプリケーションアイコンと同様に、iPhone OS 3.2では、iPadアプリケーションアイコンに次の視覚効果を自動的に追加します。

- 角丸
- ドロップシャドウ
- 反射光

アイコンがこれらの視覚強化を活かせるように、次の条件に合う72x72ピクセルの画像をPNG形式で用意してください。

- 角が90度
- 輝きや光沢がない
- アルファ透明度を使用していない

たとえば、図 3-10に、アプリケーションによって提供されるようなとても単純なアプリケーションアイコンを示します。

**図 3-10** ホーム(Home)画面に表示される前の単純なアプリケーションアイコン



図 3-11に、iPhone OSによってホーム(Home)画面に表示される同じアイコンを示します。

図 3-11 ホーム(Home)画面に表示された単純なアプリケーションアイコン



**重要：**必ず、72×72ピクセルの領域を完全に埋める画像を作成します。画像の境界がそれよりも小さい場合、または画像内に「透過領域」を作るために透明効果を使っている場合は、アイコンが角丸の黒い背景に浮かんでいるように見えるでしょう。iPhoneではこのようなアイコンを使用しないことを強く推奨していますが、iPadでは、ユーザがホーム(Home)画面に独自に写真を表示できるため特に望ましくありません。黒い背景が見えるアイコンは、iPadのホーム(Home)画面ではとても見栄えが悪いものです。

**App Storeでの表示のために、512×512ピクセル版のアプリケーションアイコンを作成する。**このバージョンのアイコンは、開発したアプリケーションのアイコンであると即座に認識できることが重要ですが、さらにいくらか表現豊かにするとともに、より精密なものにすべきです。つまり、App Store用のアイコンを作成する際に、単純にアプリケーションアイコンを拡大するべきではありません。

## Spotlightの検索結果用アイコンを作成する

どのアプリケーションも、アプリケーション名がSpotlight検索の検索文字列に一致したときにiPhone OSが表示するアイコンを提供しなければなりません。

このアイコンは、検索結果のリストからユーザがアプリケーションを簡単に見つけられるように、アプリケーションを明確に表すものにする必要があります。それには、次の仕様に従って簡素で魅力的なアイコンを作成します。

- PNG形式を使用する
- 大きさは50×50ピクセル

**注：**このアイコンの最終的な視覚サイズは48×48ピクセルです。iPhone OSは、アートワークのそれぞれの辺を1ピクセルずつトリミングし、ドロップシャドウを追加します。アイコンを設計する際には、この点を必ず考慮してください。

## 「設定(Settings)」アイコンを作成する（「設定(Settings)」を提供する場合）

「設定(Settings)」を提供する場合、「設定(Settings)」アプリケーションでデベロッパのアプリケーションをはっきりと識別する小さなアイコンを作成する必要があります。それには、次の仕様に従って簡素で魅力的なアイコンを作成します。

- PNG形式を使用する
- 大きさは29×29ピクセル

## カスタムドキュメントアイコンを提供する

iPadアプリケーションで独自のドキュメントタイプを作成する場合、ユーザにこのタイプを示すカスタムアイコンを作成するといいいでしょう。カスタムドキュメントアイコンを提供しない場合、iPhone OSはデフォルトで、アプリケーションアイコンを使ってそれを作成します。図 3-12に、図 3-11（34 ページ）に示したアプリケーションアイコンを使用して、その外観を示します。

**図 3-12** デフォルトのカスタムドキュメントアイコンはアプリケーションアイコンを表示する



必要に応じて、アプリケーションアイコンの代わりに使う独自のア트워크をiPhone OSに提供することができます。

**重要：** カスタムドキュメントアイコンを作成する場合は、必ずこのセクションのガイドラインに従ってください。アイコンが大きすぎたり小さすぎたり、または適切にパディングされていない場合は、表示されるドキュメントアイコンの外観は優れないものになります。

カスタムアイコンは、魅力的で表現豊かな細密なものであるべきです。ユーザはアイコンをさまざまな場面で目にするため、記憶に残りやすい、そしてアプリケーションに明確に関連付けられているイメージをデザインするのが最善です。

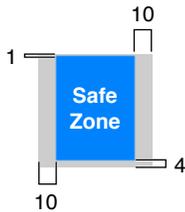
iPhone OSは、ドキュメントアイコンに、64×64ピクセルと320×320ピクセルの2つのサイズを使用します。ドキュメントアイコンが異なる場面できれいに表示されるように、両方のサイズを作成するのは良い考えです。

どちらのサイズでも全体サイズにいくらかのパディング領域が含まれているので、ア트워크用にはひと回り小さい“安全ゾーン”が残ります。重要なのは、必ずア트워크がこの安全ゾーンにうまく収まるようにすることです。さもないとア트워크がクロップされたり拡大されたりします。また、ア트워크を安全ゾーンいっぱいには描くことはできませんが、iPhone OSの追加するページ折り返し効果によって右上の角は必ず部分的に隠れます。

**64×64ピクセルのドキュメントアイコンを作成するには、次の手順に従います。**

1. 64×64画像をPNG形式で作成します。
2. 以下のマージンを追加して安全ゾーンを作成します。
  - 上部に1ピクセル
  - 下部に4ピクセル
  - 両側に10ピクセル

安全ゾーンは、次に示すハイライトされた領域のようになるはずですが。



3. カスタムアートワークを44x59ピクセルの安全ゾーン内に配置します。アートワークは中央に置いたり、中央からずらしたりできます。また、安全ゾーン全体を埋めたりすることもできます（ただし、ページ折り返し効果によって、安全ゾーンの右上角にあたるアートワークの一部領域が隠れます）。

たとえば、図3-13の左側に示す画像のようなアイコンを用意すると、iPhone OSは右側に示す画像のようなドキュメントアイコンを作成します。

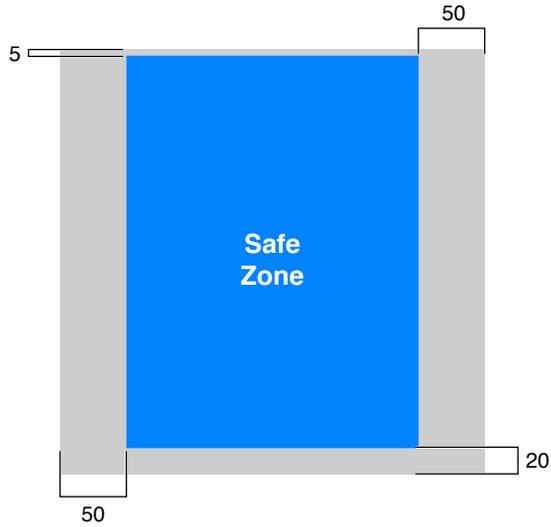
図 3-13 64x64ピクセルのドキュメントアイコンの処理の前後



320x320ピクセルのドキュメントアイコンを作成するには、次の手順に従います。

1. 320x320画像をPNG形式で作成します。
2. 以下のマージンを追加して安全ゾーンを作成します。
  - 上部に5ピクセル
  - 下部に20ピクセル
  - 両側に50ピクセル

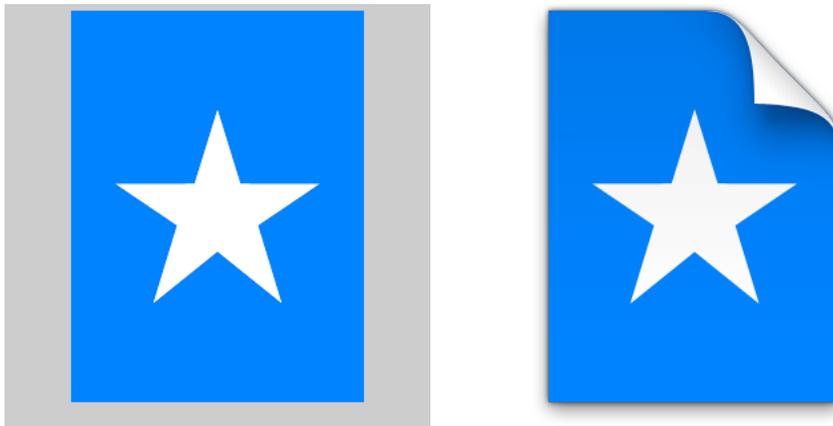
安全ゾーンは、次に示すハイライトされた領域のようになるはずですが。



3. カスタムアートワークを220×295ピクセルの安全ゾーン内に配置します。アートワークは中央に置いたり、中央からずらしたりできます。また、安全ゾーン全体を埋めたりすることもできます（ただし、ページ折り返し効果によって、安全ゾーンの右上角にあたるアートワークの一部領域が隠れます）。

たとえば、図3-14の左側に示す画像のようなアイコンを用意すると、iPhone OSは右側に示す画像のようなドキュメントアイコンを作成します。

図 3-14 320×320ピクセルのドキュメントアイコンの処理の前後



## それぞれの方向用に起動画像を作成する

ほとんどのiPadアプリケーションは4方向すべてで起動できる必要があるため、それぞれのために4つの専用画像を用意する必要があります。iPhoneアプリケーションの場合と同様に、起動画像はアプリケーションの最初のUIのスナップショットから必要な部分だけ整えた簡単な画像です。最初のUIの不変の要素だけを含め、（起動画像はローカライズされないため）一切のテキストを除外します。

起動画像はそれぞれをPNG形式にします。また、デバイス画面の向きのサイズ（1024×768ピクセル、または768×1024ピクセル）に合わせます。起動画像のファイル名の付け方、およびアプリケーションバンドルへの追加方法の詳細については、『*iPad Programming Guide*』の「Providing Launch Images for Different Orientations」を参照してください。起動画像をどれほど簡単にするべきかを示すいくつかの例については、『*iPhone Human Interface Guidelines*』の「Launch Images」を参照してください。iPhoneアプリケーションの起動画像をいくつか紹介しています。

## 確立されている原則に従う

iPadアプリケーションの開発では、『*iPhone Human Interface Guidelines*』のガイドラインからはずれないようにしてください。『*iPhone Human Interface Guidelines*』で説明しているほとんどの情報はiPadアプリケーションの設計にも当てはまります。したがって、この文書で説明していないタスク、対話、UI要素の原則とガイドラインについては、『*iPhone Human Interface Guidelines*』を参照してください。

# iPad UI要素ガイドライン

---

iPhone OS 3.2で導入された新しいUI要素および既存のUI要素の新しい動作を使用する場合は、このガイドラインに従ってください。この章で取り上げていない既存要素の使いかたを学ぶ必要がある場合は、『*iPhone Human Interface Guidelines*』の関連するセクションを参照してください。

## バー

ステータスバー、Navigation Bar、Tab Bar、およびToolbarは、アプリケーションで特別に定義された外観と動作を持つビューです。iPadアプリケーションのバーの外観と動作は、iPhoneアプリケーションのバーとほぼ同じですがいくつか異なる点があります。

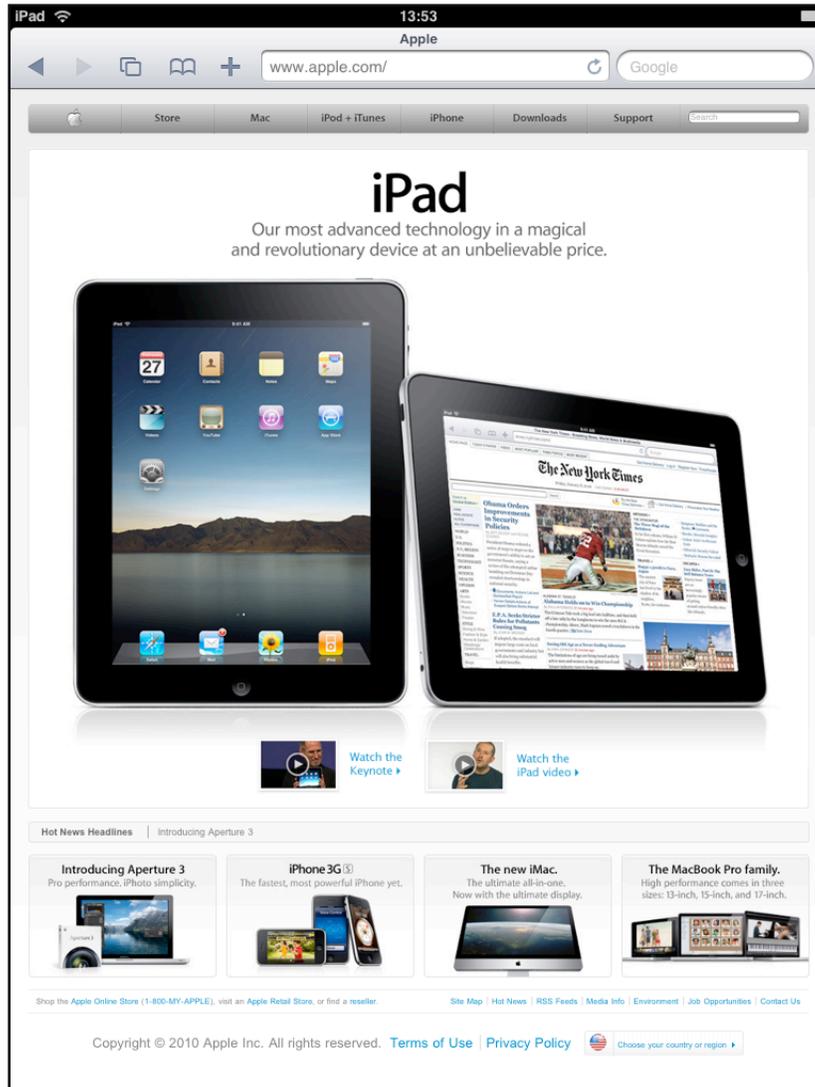
### ステータスバー

---

ステータスバーは、デバイス画面の上端に表示され（どの向きの場合でも）、ネットワーク接続、時刻、バッテリー残量など、ユーザが必要な情報を示します。

開発しているアプリケーションがゲームやフルスクリーンのメディア閲覧用アプリケーションでない場合は、ステータスバーを非表示にする前にもう一度よく考えてください。ステータスバーは画面のほんの一部しか使わないため、ほとんどのiPadアプリケーションでは、必要以上のスペースを確保するためにステータスバーを非表示にする必要はありません。iPadでは、ステータスバーの外観が控えめなので、ユーザの注意を引くためにアプリケーションと張り合うことはありません。図4-1を見て分かるように、小さなステータスバーと、アプリケーションの上部のバーの少し丸みを帯びた角が相まって、ステータスバーをデバイスの背景の一部のように見せています。

図 4-1 ステータスバーはiPadの背景に溶け込んでいる



ユーザがフルスクリーンのメディアを見ている間はステータスバー（およびその他すべてのアプリケーションUI）を非表示にすることを検討します。ステータスバーを非表示にする場合は、必ず、ユーザが1回のタップでステータスバー（および適切なアプリケーションUI）を再表示できるようにします。

ゲームではステータスバーを常に非表示にすることがありますが、この設計の決定による影響を認識しておく必要があります。ステータスバーを常に非表示にするということは、ユーザが、たとえばデバイスを充電する必要があるかどうかを確認するために、アプリケーションを終了しなければならないということです。

## Navigation Bar

**Navigation Bar**は、アプリケーション画面やビューの上端に表示されます。**Navigation Bar**には通常、現在のビューのタイトルを表示し、ナビゲーションコントロールに加え、必要に応じてビューのコンテンツを管理するコントロールを含めることができます。図 4-2に「メール(Mail)」の**Navigation Bar**を示します。

図 4-2 Navigation Barにはナビゲーションコントロールおよびその他のコントロールを含められる



**注：** **Navigation Bar**は一部のiPadアプリケーションで使用されますが、iPhoneアプリケーションでの使用ほど一般的ではありません。iPadアプリケーションで**Navigation Bar**の使用を検討する際には、必ず「[情報階層をフラットにする](#)」（19 ページ）を読んでください。

iPadアプリケーションでは、**Navigation Bar**を以下の場所で使用できます。

- Split Viewのどちらかのペイン
- Popover
- モーダルビュー
- フルスクリーンのアプリケーションビュー（ただし、iPadアプリケーションではこの使いかたはまれ）

ユーザが情報階層をドリルダウンできるようにする必要がある場合は、**Navigation Bar**を使用します。アプリケーションの最上位レベル、または個別のビュー内（タブ、Split Viewペイン、Popoverなど）で、**Navigation Bar**を使用できます。たとえば次のように使います。

- 「設定(Settings)」では、Split Viewの右のペインで**Navigation Bar**を使用し、ユーザが左のペインで選択したアプリケーションや機能に関連する設定をドリルダウンしやすくしています。
- 「カレンダー(Calendar)」では、「イベントを追加(Add Event)」Popoverで**Navigation Bar**を使用し、Popover内の一連の詳細画面を整理しています。一部の値を入力するためにユーザは新しいPopover画面に移動しますが、ユーザは元のPopover内にとどまります。
- iTunesでは**Navigation Bar**を使用して、コンテンツの一部のタブでユーザがコンテンツをドリルダウンできるようにしています。

**Navigation Bar**のタイトルには現在のビューのタイトルを使用します。ユーザが新しいレベルに移動するときは、次の2つのことを行わなければなりません。

- バーのタイトルを新しいレベルのタイトルに変更する。
- 戻るボタンをタイトルの左に表示し、戻るボタンには前のレベルのタイトルを示すラベルを付ける。

コントロールのより大きなセットを提供する必要がある場合、または移動を可能にする必要がない場合は、**Navigation Bar**の代わりに**Toolbar**を使用します。

アプリケーションのトップレベルのNavigation BarにSegmented Controlを入れることを検討します。これは、Segmented Controlによって情報階層をフラットにでき、ユーザが探しているものを見つけやすくできる場合に特に役立ちます。Navigation BarでSegmented Controlを使用する場合、配下の情報階層では戻るボタンに正確なタイトルを確実に選ぶようにしてください（使用方法のガイドラインについては、“[Segmented Control](#)”（51 ページ）を参照してください）。

スペースは十分にあるかもしれませんが、Navigation Barにコントロールを追加し過ぎないようにします。Navigation Barには、ビューの現在のタイトル以外には、戻るボタンとビューのコンテンツを管理するコントロール1つしか含めるべきではありません。そうせずに、Navigation BarでSegmented Controlを使用する場合は、バーにはタイトルを表示するべきではなく、またほかのコントロールを含めるべきではありません。

Navigation Barでは、ボーダー付きスタイルのコントロールのみを使用します。Navigation Barにプレーンな（ボーダーなしの）コントロールを配置すると、自動的にボーダー付きスタイルに変換されます。

文書化されている用途に従って、システムに用意されているボタンを使用してください。詳細については、『*iPhone Human Interface Guidelines*』の「Standard Buttons for Use in Toolbars and Navigation Bars」を参照してください。独自のNavigation Barコントロールを作ることにした場合は、設計方法に関するアドバイスについて『*iPhone Human Interface Guidelines*』の「Icons for Navigation Bars, Toolbars, and Tab Bars」を参照してください。

回転した際にNavigation Barの高さや透過性が変わらないことに注意してください。この動作は、iPhoneアプリケーションでのNavigation Barの動作と異なります。

必要に応じて、Navigation Barの色や透過性を指定します。Navigation Barをアプリケーションの全体的な外観に合わせたい場合は、カスタムの色を指定できます。ユーザをNavigation Barの背後にあるコンテンツにもっと注目させたい場合は、Navigation Barを半透明にできます。Navigation Barをこのようにカスタマイズする場合は、アプリケーションのほかの部分と外観が調和するようにしてください。

## Tab Bar

Tab Barは通常、アプリケーション画面の下端に表示されます。ユーザはこのTab Barを使って、異なるサブタスクや、ビュー、モードの間を切り替えることができます。図4-3に、iTunesのTab Barを示します。

図 4-3 Tab Barはサブタスク、ビュー、モードを切り替える



一般的に、Tab Barはアプリケーションレベルで情報を整理するために使います。Tab Barを使って情報階層をフラットにしたり、同じレベルの複数の情報カテゴリやモードに一度にアクセスできるため、Tab Barはアプリケーションのメインビューでの使用に適しています。

限られた状況において、Split ViewペインやPopoverでのTab Barの使用が理にかなう場合があります。それは、ビュー内でタブがコンテンツを切り替えたりフィルタリングしたりする場合です。しかし多くの場合、Segmented Controlの外観の方がPopoverやSplit Viewペインの外観と調和するため、PopoverやSplit Viewペインの下端ではSegmented Controlの方が適しています（Segmented Controlの使いかたの詳細については、“[Segmented Control](#)”（51 ページ）を参照してください）。

**Tab Barにたくさんのタブを詰め込み過ぎることは避けます。** Tab Barのタブを増やし過ぎると、目的のタブをタップすることが物理的に難しくなります。また、表示するタブを1つ増やすごとに、アプリケーションの複雑さが増します。一般的には、メインビューやSplit Viewの右ペインでは、タブの数は7つ程度に制限するように努めます。PopoverやSplit Viewの左ペインでは、最大5つ程度のタブが適切です。

**「その他(More)」タブの作成は避けます。** iPadでは、追加タブの一覧のみを表示する画面は、スペースをうまく活用していないということです。

**すべての向きで同じタブを表示します。** 縦向きでは、推奨する7つのタブが画面の幅にちょうど収まります。横向きでは、同じタブを画面の幅に対して中央に配置するべきです。このガイドランスは、Split Viewペイン内やPopover内でTab Barを使用する場合にも当てはまります。たとえば、デバイスが縦向きのときにPopover内でタブバーを使用する場合、同じタブ群を横向きのときのSplit Viewの左ペインで表示するとうまく機能します。

タブやタブの間隔を変えないようにすれば、アプリケーションの視覚的な安定性が高まります。図4-4に示すように、iTunesでは、横向きの場合に、縦向きの場合とまったく同じタブ（図4-3を参照）を表示します。

図 4-4 横向きの場合のTab Barに、縦向きの場合と同じ数のタブを表示する



文書化されている用途に従って、システムに用意されているタブアイコンを使用します。詳細については、『*iPhone Human Interface Guidelines*』の「Standard Icons for Use in Tab Bars」を参照してください。独自のタブアイコンを作成にした場合は、設計方法に関するアドバイスについて『*iPhone Human Interface Guidelines*』の「Icons for Navigation Bars, Toolbars, and Tab Bars」を参照してください。

デバイスの向きにかかわらず、Tab Barの色や不透明度、高さは変わらないという点に注意してください。この動作は、iPhoneアプリケーションでの動作と同じです。

## Toolbar

Toolbarは通常、画面やビューの上端に表示されますが、下端に表示することもできます。Toolbarには、画面やビューのオブジェクトに関連したアクションを実行するコントロールを含めます。図4-5に「マップ(Maps)」のToolbarを示します。

図 4-5 Toolbarは画面やビューのオブジェクトに基づいて動作するコントロールを提供する



Toolbarを使用して、頻繁に使用する、現在のコンテキストに合ったコマンドを、ユーザが選択できるようにします。また、ユーザがアプリケーションデータに違った視点でアクセスしたり、異なるアプリケーションモードにアクセスしたりできるように、Segmented ControlをToolbarに追加することが可能です（使いかたのガイドラインについては、「[Segmented Control](#)」（51 ページ）を参照してください）。

各Toolbar項目に、タップするターゲット領域を少なくとも44×44ピクセルに保ちます。Toolbar項目を詰め込み、項目の間隔が狭くなり過ぎると、ユーザが目的の項目をタップすることが難しくなります。

文書化されている用途に従って、システムに用意されているToolbar項目を使用します。詳細については、『*iPhone Human Interface Guidelines*』の「Standard Buttons for Use in Toolbars and Navigation Bars」を参照してください。独自のToolbar項目を作成にした場合は、設計方法に関するアドバイスについて『*iPhone Human Interface Guidelines*』の「Icons for Navigation Bars, Toolbars, and Tab Bars」を参照してください。

プレーンスタイル（ボーダーなし）とボーダー付きスタイルのToolbar項目を、同じToolbarに混在させるのは避けるようにします。Toolbarではどちらのスタイルも使用できますが、通常、両方を混在させると見栄えが良くありません。

## コンテンツビュー

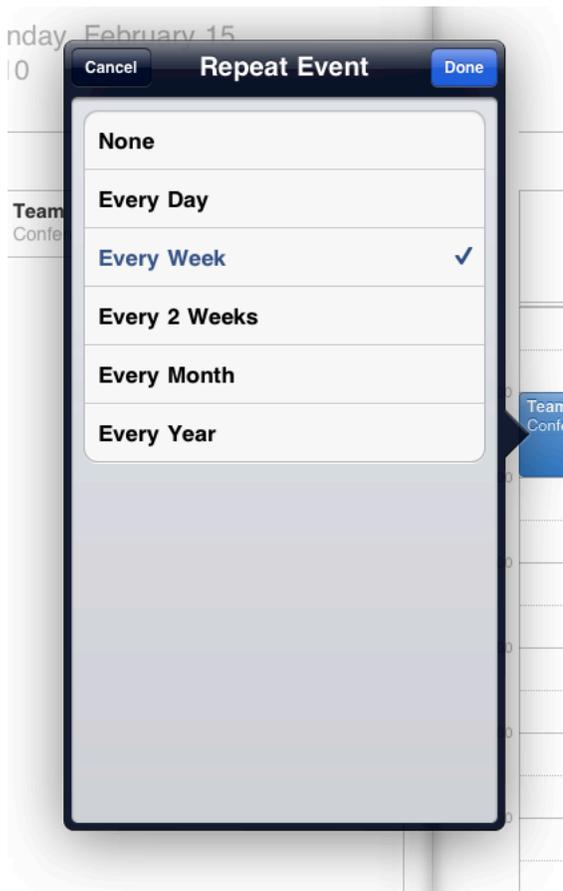
既存のImage View、Map View、Table View、Text View、Web Viewに加え、iPhone OS 3.2では、コンテンツを管理する2つの新しいビューを導入しました。**Popover**と**Split View**です。Text Viewの新しい機能を除くと、その他の既存のコンテンツビューの動作は以前のバージョンのiPhone OSでの動作と同じです。

### Popover

---

**Popover**は、コントロールまたは画面上の領域をタップしたときに表示される一時的なビューです。図 4-6に「カレンダー(Calendar)」のPopoverを示します。

図 4-6 Popoverは一時的なビューで選択肢または機能を提示する



**重要：** PopoverはiPadアプリケーションでのみ利用できます。

Popoverにはさまざまなオブジェクトやビューを含められます。たとえば、次のようなものを含められます。

- Table View、Image View、Map View、Text View、Web View、カスタムビュー
- Navigation Bar、Toolbar、Tab Bar
- 現在のアプリケーションビューのオブジェクトを対象に動作するコントロールやオブジェクト

Popoverを使って以下を行えます。

- 焦点を当てた（または選択した）オブジェクトに関連する追加情報や項目リストを表示する。
- Split Viewベースのアプリケーションが縦向きの際に、左ペインのコンテンツを表示する。この場合、Popoverを表示する適切な名前を付けたボタンを必ず用意します。ボタンは、できれば画面上部のNavigation BarまたはToolbarに配置します。
- 画面に表示されているものと密接に関連するオプションの短いリストを示すAction Sheetを表示する。

**注：**Popoverには常に矢印が表示されます。Popoverの背景やボーダーの外観を変更することはできません。

**“Popoverを閉じる(dismiss popover)”ボタンを付けるのは避けます。** Popoverは、不要になれば自動的に閉じるようにするべきです。たとえば次のようにします。

- Popoverの機能が、メインビューに影響する一連の選択肢や項目を示すだけの場合、ユーザが選択を行ったらPopoverをすぐに閉じるべきです。この動作は、コンピュータアプリケーションのメニューの動作にとってもよく似ています。この動作は、Action Sheetのみが表示されるPopoverにも当てはまることに留意してください。ユーザがAction Sheetのボタンをタップしたら、Popoverはすぐに閉じるべきです。

場合によっては、メインビューに影響する項目を含みながらも、ユーザが選択したときに閉じないPopoverを提供することが理にかなっている場合もあります。Popoverにインスペクタを実装する場合はこのようにすると良いでしょう。ユーザが、追加の選択を行ったり現在の選択項目の属性を変えたい場合があるためです。

メニューやインスペクタ機能を提供するPopoverは、Popoverの境界の外側をタップしたときに閉じるべきです。選択肢のメニューを示すPopoverでは、このジェスチャは、ユーザが選択しないことにした（このためメインビューへの影響はない）、という意味です。Action Sheetを含むPopoverでは、このジェスチャは「キャンセル(Cancel)」ボタンをタップする代わりになります。

- タスクを可能にするPopoverの場合、タスクを完了または中止し、かつ同時にPopoverを閉じるボタンを表示させるのが妥当なことがあります。一般的に、編集タスクを可能にするPopoverには、「完了(Done)」ボタンや「キャンセル(Cancel)」ボタンを表示します。これらのボタンは、編集環境にいることをユーザに思い出させるのに役立つとともに、入力を明示的に保持したり破棄したりする手段を提供します。ユーザがどちらかのボタンをタップしたら、Popoverは閉じるべきです。

アプリケーションで理にかなうのであれば、このようなPopoverのボーダーの外側をユーザがタップしたときにPopoverを閉じないようにすることが可能です。ユーザがタスクを終了する（または明示的に破棄する）ことが重要な場合には、これは良い方法かもしれませんが。そうしない場合は、ユーザがPopoverのボーダーの外側をタップしたときに、「完了(Done)」をタップした場合と同じようにユーザの入力を保存するべきです。

**一般に、ユーザがPopoverのボーダーの外側をタップしたときはユーザの作業を保存します。**これは、Popoverは明示的に閉じる必要がないためユーザが誤ってPopoverを閉じてしまう可能性があるためです。「キャンセル(Cancel)」ボタンをタップしたときのみ、ユーザの作業を破棄するようにします。

**Popoverの矢印は、可能な限り直接、Popoverを表示した要素を指すようにします。**これは、Popoverがどこから表示され、どのタスクやオブジェクトに関連しているのかを、ユーザが覚えておくのに役立ちます。

**ユーザがPopoverの背後のアプリケーションコンテンツを見なくてもPopoverを使えるようにします。** Popoverは背後のコンテンツを覆い隠し、ユーザはPopoverを別の場所にドラッグすることはできません。

**一度に画面上に表示するPopoverは必ず1つだけにします。** Popoverはユーザのアクションに密接に関連しているため、画面上に複数のPopoverがあるとユーザを混乱させることになるでしょう。

**可能であれば、1回のタップで1つのPopoverを閉じると同時に新しいPopoverを開けるようにします。** この動作は、複数の異なるバーボタンがそれぞれ別のPopoverを開く場合に特に適しています。その理由は、ユーザが何度もタップする必要がなくなるためです。

**Popoverを大きくし過ぎるのは避けます。** Popoverが画面全体を覆うように表示させるべきではありません。Popoverはコンテンツを表示するのに十分な大きさにとどめ、かつPopoverの表示元を指すようにする必要があります。

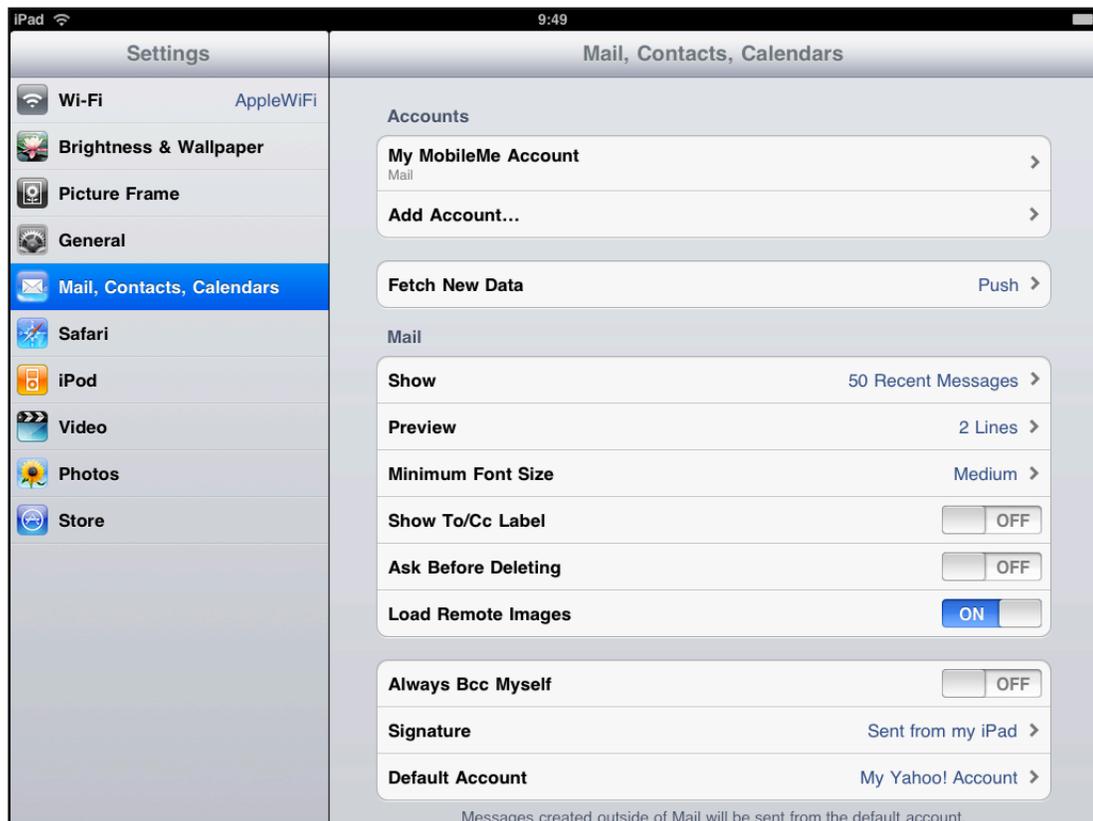
理想的にはPopoverの幅は少なくとも320ポイントにすべきですが、600ポイントより大きくすべきではありません。Popoverの高さに制限はないため、長い項目リストの表示にPopoverを使うことができます。しかし、一般に、タスクを可能にするPopoverやAction Sheetを示すPopoverでは、スクロールを避けるべきです。システムが、Popoverの高さと幅の両方を画面に合うように調整する可能性がある点に留意してください。

**適切と考えられれば、表示されている状態でPopoverのサイズを変更します。** たとえば、Popoverを使用して同じ情報の最小ビューと拡張ビューの両方を表示する場合は、Popoverのサイズを変えると良いでしょう。Popoverが表示されているときにそのサイズを変更する場合は、変更をアニメーション化するという方法が選べます。アニメーション化すると、ユーザが、古いPopoverが新しいPopoverに置き換わったと思うことを防げるため、変更をアニメーション化するというのは通常良い考えです。

## Split View

**Split View**は、2つの隣り合ったペインから成る1つのフルスクリーンビューです。左のペインはマスタペインと呼ばれることが多く、右のペインは詳細ペインと呼ばれることが多いのですが、Split Viewは、マスタと詳細から成る設計の実装に限定されるものではありません。図 4-7に、「メール (Mail)」のSplit Viewを示します。

図 4-7 Split Viewには2つの関連するビューが含まれる



**重要：** Split ViewはiPadアプリケーションでのみ利用できます。

通常、Split Viewを使い、左ペインに変わらない情報を表示し、右ペインに関連する詳細や下位の情報を表示します。左ペインで項目を選択すると、右ペインにその項目に関連する情報が表示されず（このように動作させるコードはデベロッパが実装します）。Split Viewを使用している例としては、「メール(Mail)」と「設定(Settings)」の2つがあります。

- 横向きのとき、「メール(Mail)」は左ペインにユーザのアカウントとメールボックスの階層を表示し、右ペインに選択されたメッセージを表示します。ユーザは左ペインでコンテンツをドリルダウンし、右ペインで最も詳細な情報を見ます。
- すべての方向において、「設定(Settings)」は左ペインに最上位のデバイスとアプリケーションカテゴリを表示し、右ペインに個別の設定を表示します。ユーザは左ペインでカテゴリを選択し、必要に応じて右ペインで関連する設定をドリルダウンします。

両方のペインに、以下のようなさまざまなオブジェクトやビューを含められます。

- Table View、Image View、Map View、Text View、Web View、カスタムビュー
- Navigation Bar、Toolbar、Tab Bar（ただし、Split Viewの両方のペインに同時にNavigation Barを表示しても意味はありません。）

一般的に、デバイスが横向きのときにアプリケーションでSplit Viewを使用する場合、縦向きに回転したときには、左ペインの内容をPopoverに表示します。しかし、このパターンに従わなければならないというわけではありません。アプリケーションで妥当であれば、どの方向でもビューを横に並べて表示するようにUIを設計できます。

**左ペインよりも狭い右ペインを作ることは避けます。**左ペインの幅は、どの向きの場合でも320ポイントで固定です。右ペインの幅は任意ですが、320ポイントよりも狭くすると見栄えが良くありません。

**一般に、左ペインでは現在の選択を持続的に示します。**この動作は、左ペインの項目と右ペインのコンテンツの関係をユーザが理解するのに役立ちます。右ペインのコンテンツは変わりますが、左ペインで選択した項目に常に関連し続けるべきであるため、これは重要です。

## Text View

テキストのレイアウト計算にCore Textを使用したり、結果の描画にUITextViewオブジェクトを使用したりすることは避けます。このような使いかたはサポートされていません。Core Textの機能は、フル機能を備えたワープロアプリケーションの実装に役立つことを目的としています。よりシンプルなテキスト処理機能を必要とするアプリケーションの大半には、Core Textは不要であるか適しません。

iPadアプリケーションではデベロッパ独自のフォントを使用できます。使用するフォントをアプリケーションバンドルに含め、Info.plistファイルのUIAppFontsキーを使用して、フォントを指定します。このファイルの詳細については『*iPhone Application Programming Guide*』の「The Application Bundle」を、iPadに固有のガイドラインについては『*iPad Programming Guide*』の「Updating Your Info.plist Settings」を参照してください。

.ttfフォントや.otfフォントはiPadで使用できますが、従来のMac OSフォント（つまり、Classicフォントスーツケース）は、iPadでは使用できません。

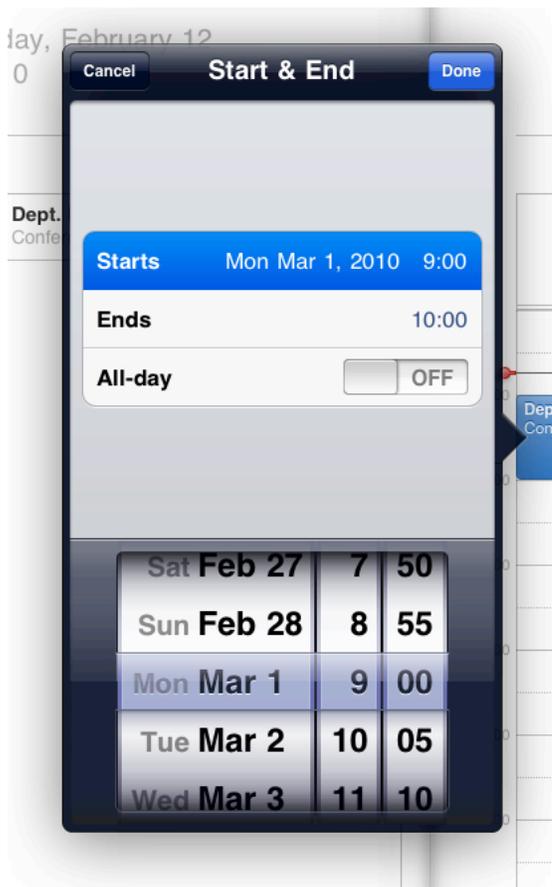
## コントロール

iPhone OSの過去のバージョンで導入されたUIKitコントロールはすべてiPadアプリケーションで利用できます。わずかな違いはいくつかありますが、コントロールの外観と動作はデベロッパやユーザーが期待するとおりのものです。

### 日付と時刻のピッカー、ピッカー

日付と時刻のピッカーを使うと、日付など複数の部分から成る値を選択できます。ピッカーは日付と時刻のピッカーの汎用版です。ユーザはピッカーを使用して、単一の要素で構成されるひとつの値から特定の値を選択できます。図 4-8に、「カレンダー(Calendar)」の日付と時刻のピッカーを示します。

図 4-8 日付と時刻のピッカーによって複数の部分から成る値の選択が簡単になる



ピッカーまたは日付と時刻のピッカーはPopover内にもみ表示します。この配置は、iPhoneアプリケーションで推奨する配置とは異なります。このコントロールで行えることやコントロールのカスタマイズ方法の詳細については、『iPhone Human Interface Guidelines』の「Date and Time Pickers」および『iPhone Human Interface Guidelines』の「Pickers」を参照してください。

## Infoボタン

---

「Info」ボタンを使って画面全体をフリップするのは避けます。代わりに、関連情報や詳細情報を含む拡張ビューにアクセスできることをユーザに示すために、「Info」ボタンを使うと良いでしょう。この使いかたは、iPhoneでの「Info」ボタンのもっとも一般的な使いかたとは異なります。iPhoneでは一般的に、ユーティリティアプリケーションの画面をフリップして設定オプションを表示するために使います。

## ページインジケータ

---

iPadは画面が大きいため、ページインジケータコントロールの必要性が低くなります。iPhoneアプリケーションでページインジケータを使用した場合は、コンテンツを複数の同列の画面ではなく1つの画面に表示する方法を検討してください。

## Search Bar

---

**Search Bar**は、アプリケーションが検索の入力として使用できるユーザからのテキストを受け取るフィールドです。ユーザがSearch Barをタップすると、キーボードが表示されます。ユーザが検索用語の入力を終わると、その入力はアプリケーション固有の方法で処理されます。図 4-9に、「メール(Mail)」におけるPopoverのSearch Barを示します。

図 4-9 Search Barはユーザからの入力を受け取る

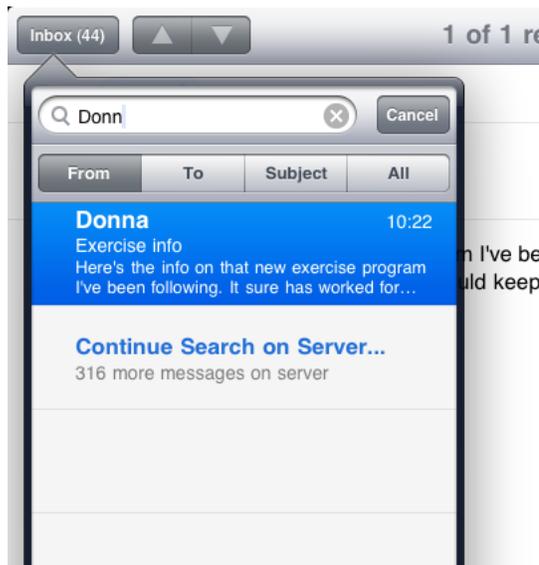


ユーザが素早く検索機能にアクセスできるように、**Navigation Bar**や**Toolbar**でSearch Barを使います。アプリケーションの主要な機能が検索の場合、iTunesと同じように検索タブを提供すると良いでしょう。

ユーザが前回の検索結果を取得したり変更したりできるように、**検索結果ボタン**を表示します。ユーザはこのボタンをタップして、Popoverで検索結果のリストを見ます。アプリケーションにとって意味があれば、ユーザによる文字入力に合わせてリストを絞り込むことができます（あるいはオートコンプリートや提案機能を提供することができます）。Search Barに検索結果ボタンを表示する方法については『*UISearchBar Class Reference*』を参照してください。

ユーザが**検索範囲を変更**できるように、**Scope Bar**の使用を検討します。iPadアプリケーションでは、図 4-10に示すように、サブビュー内（たとえば、Split Viewの左ペインやPopover内）のSearch Barの下にScope Barを配置できます。

図 4-10 Scope Barはユーザが検索を絞り込むのに役立つ



## Segmented Control

**Segmented Control**は直線形のセグメントセットです。それぞれのセグメントは、異なるビューを表示できるボタンとして機能します。図 4-11に、iTunes ToolbarのSegmented Controlを示します。

図 4-11 Segmented Controlを使用して異なる視点やビューへのアクセスを提供できる

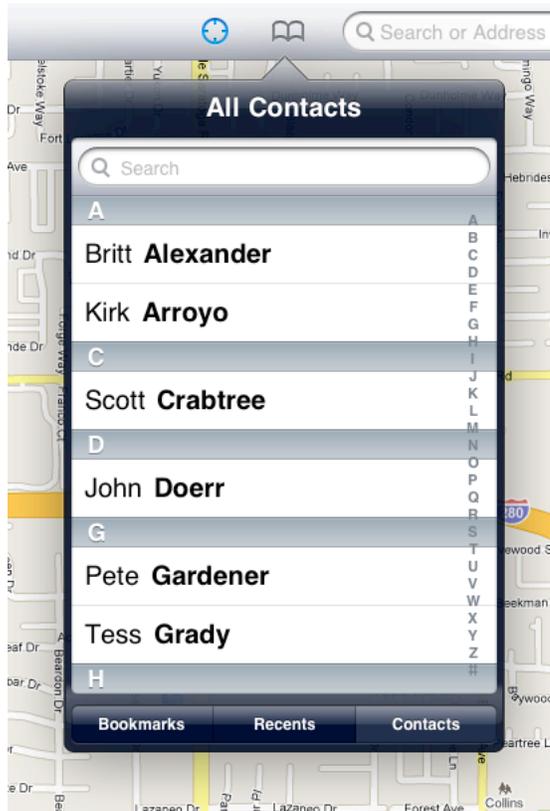


アプリケーションで異なる視点やモードを提供するために、上端のToolbarでSegmented Controlを使います。Segmented Controlを画面の上部に配置すると、ユーザはそのコントロールをアプリケーションレベルでデータカテゴリやモードを管理する手段として認識する傾向があります。たとえば、「マップ(Maps)」では上端のToolbarでSegmented Controlを使い、ユーザが検索モードと道順モードを切り替えられるようにしています。

ビューで異なる視点やモードを提供する場合には、下端のToolbarでSegmented Controlを使います。Segmented Controlをビューの下部に配置すると、ユーザはそのコントロールをビューレベルでデータカテゴリやモードを管理する手段として認識する傾向があります。たとえば、iPodアプリケーションは下端のToolbarでSegmented Controlを使い、ユーザがいろいろな方法でライブラリをグループ化できるようにしています。

ビュー固有のSegmented Controlのもう1つの例は、図 4-12に示す、「マップ(Maps)」のブックマークのPopoverにあります。Segmented ControlがPopoverに含まれているため、このSegmented ControlはPopoverのコンテンツを対象に動作し、アプリケーションのほかの情報をフィルタしたりモードを変更したりしないことがユーザに分かります。PopoverやSplit Viewペイン内でモード切り替えやフィルタリング機能を提供する必要がある場合は、Tab BarではなくSegmented Controlを使うことを検討します。

図 4-12 Segmented ControlはPopoverでは情報をフィルタリングしたりモードを切り替えたりするのに有効



## Alert、Action Sheet、およびモーダルビュー

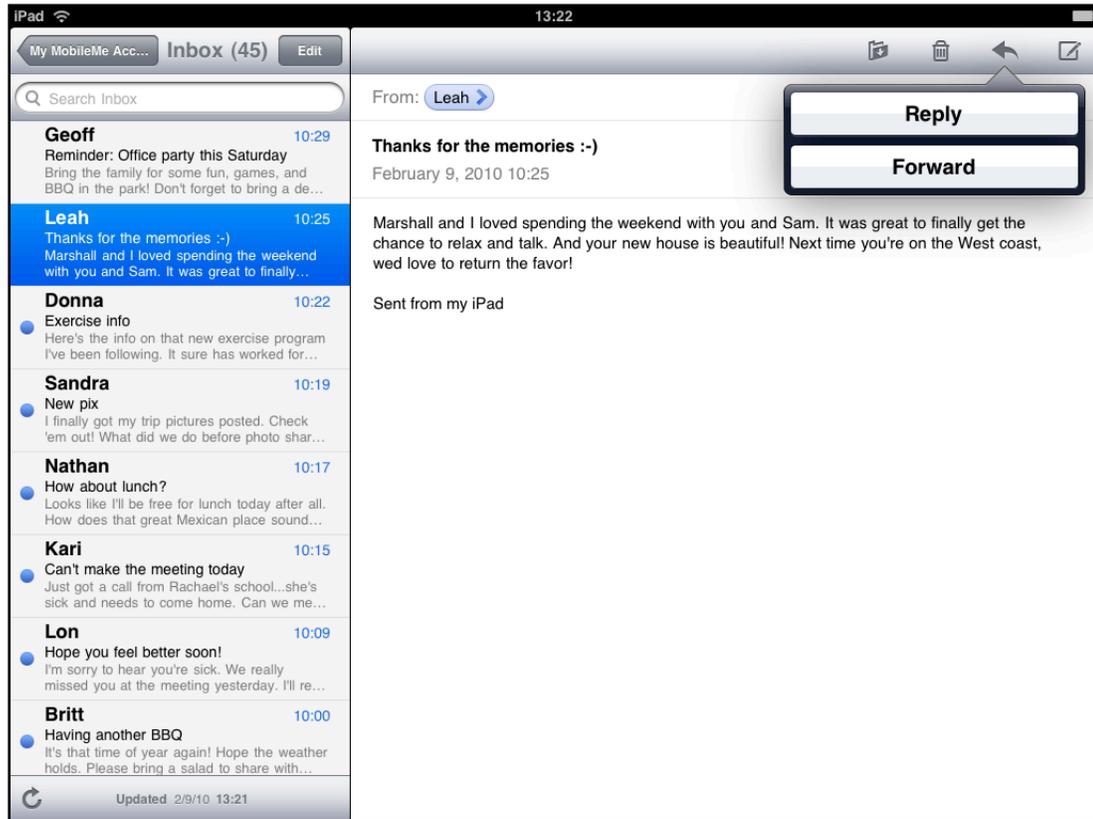
Action Sheet、Alert、およびモーダルビューは、ユーザに注意を呼びかける場合や、アプリケーションが追加の選択肢や機能をユーザに提示する必要がある場合に表示する一時的なビューです。これらのビューのいずれかが画面に表示されている間は、ユーザはアプリケーションとやり取りできません。

### Action Sheet

**Action Sheet**には、ユーザが開始したタスクに関連する一連の選択肢を表示します。

iPadアプリケーションでは、Action SheetはPopover内に表示され、フルスクリーンの幅になることはありません。図 4-13に、ユーザが「メール(Mail)」の「返信(Reply)」ボタンをタップしたときに表示されるAction Sheetを示します。

図 4-13 Action SheetはPopover内に表示される



Action Sheetは、アニメーション付きでもアニメーションなしでも表示できます。

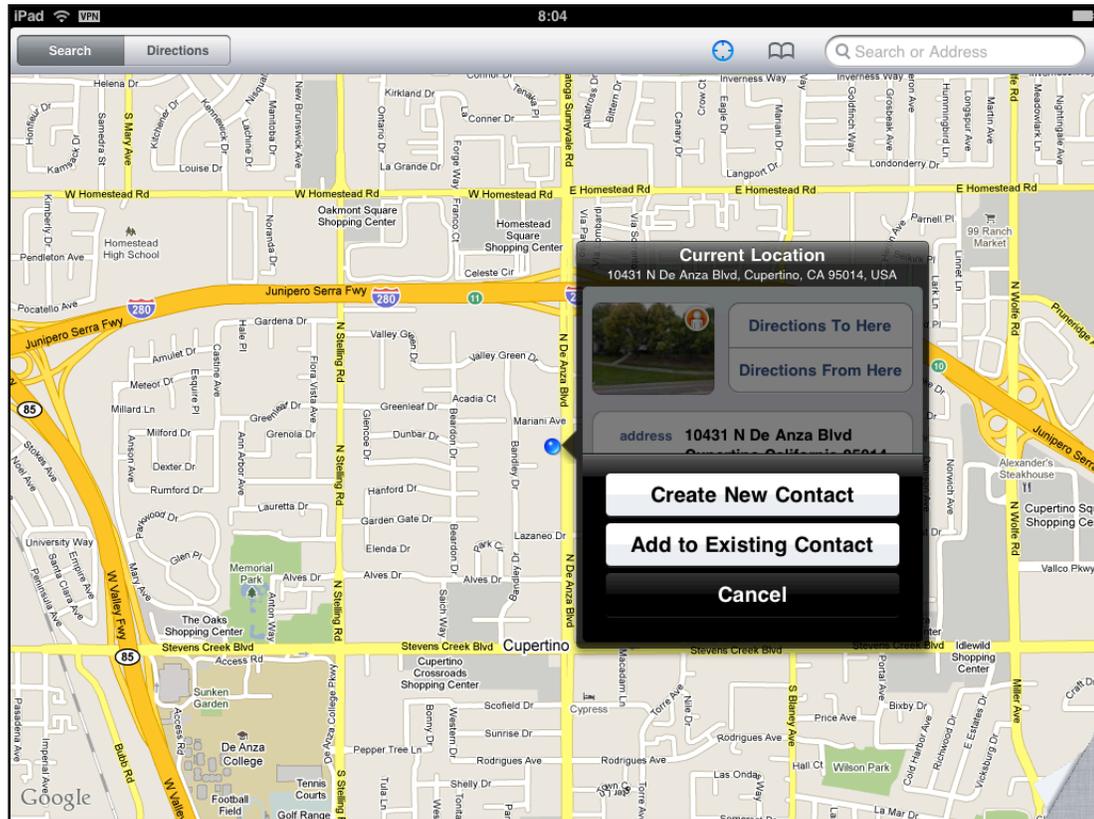
- **アニメーションを使わない場合、Action SheetとPopoverは同時に表示されます。** ユーザがPopoverの外で開始したタスクに関連する選択肢を提供する場合には、アニメーションを使わずにAction Sheetを表示します。この方法でAction Sheetを表示した場合、Popoverの矢印は、ユーザがタスクを開始するためにタップしたコントロールや領域を指します。

ユーザは、選択肢のいずれも選択せずにPopoverの外側をタップすることでAction Sheetを閉じることができるため、「キャンセル(Cancel)」ボタンは付けません(図4-13のAction Sheetには「キャンセル(Cancel)」ボタンがないことに注目してください)。

- **アニメーションを付けると、Action Sheetは開いているPopoverのコンテンツの上を、上方向にスライドします。** 開いているPopover内から開始するタスクに関連する選択肢を表示する場合には、アニメーションを使用します。

アニメーション化したAction Sheetには「キャンセル(Cancel)」ボタンを付けるべきです。これは、ユーザがPopoverを閉じずにAction Sheetを閉じられるようにするためです。図4-14に、「マップ(Maps)」の場所のPopoverで表示されるAction Sheetを示します。

図 4-14 アニメーション化されたAction Sheetには一般的に「キャンセル(Cancel)」ボタンが付いている



## Alert

Alert（警告）は、iPhoneの場合と同じようにiPadで動作します（Alertをどのような場合に使用すべきか、また、適切なAlert体験を設計する方法については、『*iPhone Human Interface Guidelines*』の「Using Alerts」を参照してください）。

簡単に言うと、Alertは、アプリケーションやデバイスの使用に影響する重要な情報をユーザに提供するために使うべきです。Alertが必要となることはまれですが、以下のような場合に最適です。

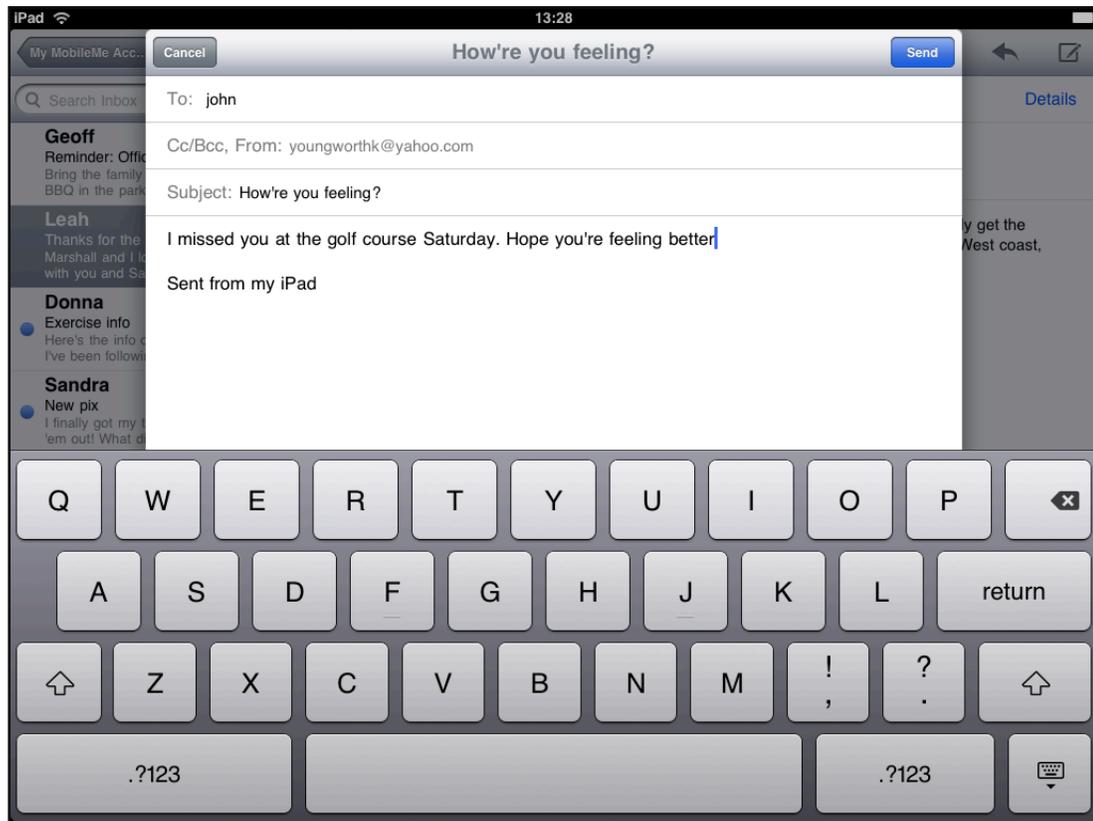
- 問題を示し、ユーザに問題を処理する方法を選ばせる
- 危険な可能性のある結果を受容または拒否する機会をユーザに提供する

なんらかの情報、特にアプリケーションの標準機能に関する情報がより目につくようにしたいだけであれば、Alertの使用は避けるべきです。代わりに、アプリケーションのスタイルに調和し、目を引く情報表示の方法を設計すべきです。

## モーダルビュー

**モーダルビュー**（つまり、モーダルモードで表示されるビュー）では、現在のタスクやワークフローのコンテキスト内で自己完結した機能を提供します。図4-15に「メール(Mail)」のモーダルビューを示します。

図 4-15 モーダルビューでは自己完結するタスクを表示する



モーダルビューは、現在のタスクおよびiPadアプリケーションの視覚的なスタイルに合うスタイルで表示できます。

- **フルスクリーン**。画面全体を覆います。このスタイルは、ユーザがモーダルビューのコンテキスト内で完結できる、複雑になり得るタスクを表示するのに適しています。たとえば、iPodアプリケーションでは、Geniusプレイリスト(Genius Playlist)作成タスクに使用しています。
- **Page Sheet**。幅は768ポイントに固定されています。シートの高さは画面の現在の高さです。縦向きの場合、モーダルビューは画面全体を覆います。横向きの場合、モーダルビューの両側に見える画面は、ユーザが操作しないようにグレイアウトされます。たとえば「メール(Mail)」では、メッセージ作成タスクにこのスタイルを使用します（図 4-15に示します）。
- **Form Sheet**。サイズは540×620ポイントに固定されており、画面の中央に配置されます。モーダルビューの外側の画面が見えている部分は、ユーザが操作しないようにグレイアウトされます。デバイスが横向きで、キーボードが表示されるときは、Form Sheetは上に移動し、ステータスバーのすぐ下までできます。このスタイルは、ユーザから構造化された情報を集めるときに有効です。

- **現在のコンテキスト。** 親ビューと同じサイズを使用します。このスタイルは、Split Viewペイン、Popover、またはその他のフルスクリーンではないビューの中でモーダルビューを表示するのに適しています。

既存のトランジションスタイル（垂直とフリップ）に加え、iPhone OSでは部分カールトランジションを導入しています。このトランジションを使用すると、現在のビューの1つの隅が巻き上がり、その下のモーダルビューを表示します。ユーザがモーダルビューを抜けると、巻き上がっていた現在のビューが元の位置に戻ります。部分カールトランジションで表示されたモーダルビュー自体は、別のモーダルビューを表示できないことに留意してください。

部分カールトランジションスタイルは、iPhoneの「マップ(Maps)」のページカールの動作に似ています。表示するモーダルビューが大きくなり、設定オプションを示すビューのようにユーザとのやり取りをさほど必要としないときは、このスタイルを使用すると良いでしょう。

ほかのモーダルビューを持てるモーダルビューを表示しようと考えている場合は、複雑なやり取りは避けてください。ガイダンスについては、「[モーダルタスクでは複雑さを制限する](#)」（28ページ）を参照してください。

## 編集メニューの追加

一般的に、編集メニューに追加する項目は、ユーザが選択している対象を編集、変更するなど、直接働きかけるものであるべきです。ユーザは、標準の編集メニューが現在のコンテキスト内のテキストやオブジェクトを対象に動作することを期待しており、カスタムメニュー項目も同じように動作するのが最善です。

**注：** 現在のコンテキスト外の方法で、選択したテキストやオブジェクトを使用するアクションを可能にする必要がある場合には、Action Sheetを使うほうが良いでしょう。たとえば、ユーザが選択している項目をほかのユーザと共有できるようにしたいなら、ユーザの選択項目の送信先となるソーシャルネットワーキングサイトを一覧にしたAction Sheetを表示すると良いでしょう。Action Sheetの使用ガイドラインの詳細については、「[Action Sheet](#)」（52ページ）を参照してください。

編集メニューにカスタム項目を追加する場合は、必ずシステムに用意されている項目のあとにまとめてリストアップします。システムに用意されている項目の間にカスタム項目を混ぜ込まないようにしてください。

カスタムメニュー項目は妥当な数にとどめます。ユーザを圧倒してしまわないように、選択肢を与えすぎないようにします。

カスタムメニュー項目には簡潔な名前を付け、そのコマンドが何をすることを正確に表していることを確認してください。一般に、項目名は、実行するアクションを表す動詞にすべきです。英語の項目名では先頭が大文字の1つの単語を使用するのが望ましいですが、短い句を使う必要がある場合は、文章のタイトルと同様の大文字の使い方をしてください（簡単に言うと、文書のタイトルと同様の大文字の使い方とは、4文字以下の冠詞、等位接続詞、および前置詞を除くすべての単語の1文字目を大文字にすることを意味します（英語の場合））。

## キーボードのカスタマイズ

iPhone OS 3.2では、システムに用意されている画面上のキーボードに代わるカスタム入力ビューを設計できます。カスタム入力ビューを提供する場合、その機能がユーザにとって明白であるようにしてください。また、作成した入力ビューのコントロールがタップできると分かるような外観にします。



# 書類の改訂履歴

---

この表は「iPadヒューマンインターフェイスガイドライン」の改訂履歴です。

日付	メモ
2010-04-03	iPadアプリケーション向けに魅力的なユーザインターフェイスを設計する方法について説明する新規文書。

## 改訂履歴

書類の改訂履歴