

7  
章

# 第一印象

—— アプリの効果的な紹介

First Impressions: INTRODUCING YOUR APP

アプリとの関係は、人間との関係と同じで第一印象に大きく左右されます。ユーザーをアプリの世界に引き込み、快適な時を過ごしてもらうためには、そのアプリに出会った瞬間に、魅力的で信頼感にあふれたアプリであること、そして簡単に使えるアプリであることを感じとってもら

らなければなりません。アプリの起動画面や紹介画面は単なる飾りだと思われがちですが、第一印象に及ぼす影響は大きいのです。アイコンに気を配らないのは、爪が汚いまま合コンに行くようなものです。「完全無視」が避けられたとしても、魅力的にはうつりません。不恰好なアイコンではアプリを試す気をなくします。合コンやデートには、きちんと髪をとかして小ぎれいな身なりで出かけましょう。この章では、どうすれば「ひと目惚れ」されるアプリを作れるかを検討します。



Photo: L. Broadwell

## 7.1 | アイコンは名刺

名刺とは予測を生み出すものです。誰かに名刺を差し出すときに渡しているのは、連絡先だけではありません。しっかりした紙か？家でインクジェットプリンタで印刷したか？派手なデザインか？色は？どんなフォントを使っているか？ロゴの画像は見栄えがするか？これらすべての見だ目次第で、名刺デザインの印象だけでなく、それを手渡した人の印象も決まります。名刺はあなた（あるいは少なくともあなたの会社）のプロとしての魅力を伝えると同時に、必要十分な情報を伝えるものでなければなりません。この薄っぺらな紙切れこそが、初めて会った人々にとってはあなたがどうい

るかを知るための数少ない手掛りなのです。これと同じことが App Store や iPhone のホーム画面に並べられるアプリのアイコンについてもいえます。

アプリに出会う人が最初に目にするのがアイコンです。App Store の検索結果で、アプリの名前よりも目立ってしまうものなのです。App Store のリストに表示されるのは、アイコン、名前、分類、そして価格です。詳細画面でもアイコンはスクリーンショットよりも上に表示されます。美しいアイコンを目にしたユーザーは、アプリ自身も同様だろうと考え、ダウンロードしようと思うでしょう。アイコンは点の集まりでしかありませんが、マーケティング面から見ると、そのサイズとは比べものにならないほどの影響力をもっています。ですからアイコンをデザインするときも十分な時間をかけましょう。自分でデザインできないなら、これだけをデザイナーに依頼することを検討してみてください。それだけの価値はあるのです。

アイコンは、ほかのアイコンといつも「肩を並べる」ことになるので、その中でも独自性をもち目



図 7-1 アイコンをデザインする際は、アプリの内容をわかりやすく表現するためにインターフェースの基調色、スタイル、テクスチャを流用する。左上から時計回りに Cellar、Deep Green、Canabalt の各アプリの画面とそのアイコン。

立つものでなければなりません。まず色について考えましょう。青や緑はアイコンとしてはあたりまえすぎるので、違う色を選んだほうがよいでしょう。自分のアプリのインターフェースからインスピレーションを得てください。このアプリにしかないといったような<sup>メタファー</sup>比喻や色使いがあるならそれをどうアイコンに反映させるか考えましょう。アイコンを見て選んだユーザーの期待にアプリが応えられれば、信頼が生まれます。そのためにはインターフェースからアイコンのデザインを借りるのもひとつの方法です。

アイコンはカラフルで楽しく、一度見たら忘れられないものにしましょう。とはいえ、謎めいたものにはしないでください。アイコンは基本的には広告であって、よい広告のすべてがそうであるように、何を売っているかが明確でなければなりません。タップしたくなるような、アプリに親しみを感じるような「なにか」を直接的に伝えるべきです。それには機能、インターフェース、名前、ブランドといったものを視覚的に表現することが考えられます。モバイル環境ではユーザーに考え込ませたり、躊躇させたりするようなことは避けましょう。アイコンは商品のパッケージに書かれているイラストだと思ってください。カワイイもの、美しいものができればそれにこしたことはありませんが、ブランドの確立という観点からもユーザーへの適切な情報提供という観点からも、ひと目で内容がわかって識別できることのほうがはるかに重要です。わかりにくい画像や、普通の人に想像がつかないようなメタファーも避けましょう。売上アップにつなげるようにします。芸術作品を作っているのではあ

りません。

Iconfactory は自社のゴージャスな雰囲気 of Ramp Champ という iPhone ゲームでの失敗を通して、この教訓を学びました。見た目は美しいけれども目立たないアイコンと、曖昧なネーミングのせいで App Store の顧客が混乱したのです。あとで考えたところアイコンが伸び悩みの原因だということに



図 7-2 Ramp Champ のアイコン (左) はユニークで記憶に残りはするが、何をやるアプリであるか想像がつきにくい。Freeverse が開発した Skee-Ball のアイコンのほうが、ゲームの内容が想像でき効果的だといえる。

思い至りました。Craig Hockenberry 社長はこう語っています。「アイコン自体はユニークで App Store でも目立っていましたが、わかりにくくメッセージがうまく伝わりませんでした。ユーザーにはわからなかったんですね。それで、タップしたくなるほどには引きつけられなかったのです。Ramp Champ を通して、アイコンはたとえそれが見た目という点からはベストなものでもなく、わかりやすくなければならぬということを学びました」。

アイコンでアプリを表現するための手法をいくつか紹介しましょう。すでに見たように、デザインに特徴をもつアプリの場合、そのデザインを彷彿とさせ、アプリ自体のプレビューのような役割を果たすものにすることが考えられます。このプレビューによってアプリの機能を示唆することも可能かもしれませんが、外観を強調することになるため、デザインが重要なアプリにふさわしいアプローチといえるでしょう。

より直接的な手法としては、アプリの機能、つまりそのアプリが何をしてくれるかに焦点を合わせるものがあります。道路標識のようにわかりやすいシンボルを用いて、何が出来るかを示します。一般的になりすぎないよう、ユニークさを保ったまま内容を説明するようなデザインにするのが難しいところです。類似のアプリが多い場合、アイコンまで似てしまわないよう注意が必要です。ToDo リスト管理用のアプリは山ほどありますが、数少ない例外を除き、そのほとんどがアイコンにチェックマークを使っています。また、写真関連アプリの開発者はカメラのレンズの画像を使うという誘惑になかなか打ち勝てないようです。機能をうまく表現しつつ、ほかのアプリと違うことがしっかりわかるアイコンを考えましょう。図 7-3 のようなアイコンは成功例といえるでしょう。



図 7-3 機能を表現したアイコンの例。左から Cha-Ching (ファイナンス)、Kineo (バラバラ漫画)、cab4me (タクシー呼び出し)、Cocktails+ (カクテルレシピ)、Lose It! (カロリー計算)、Delivery Status Touch (宅配便追跡)。

機能が抽象的すぎて、単純な絵柄では容易に説明できないようなアプリもあります。そういった場合、タイトルそのもののブランドの確立を狙ってアイコンにアプリの名前を表示する方法があります。ただしこれはリスクを伴う方法といえるでしょう。というのは、アイコンを見ても何をやるアプリか、

どのようなインタフェースをもつものかわからないのです。したがって、人を引きつけるような、質の高い、潜在的なユーザーにも安心感を与えるようなアイコン、少なくとも好奇心をそそるようなアイコンが望まれます。このアプローチでは、類似のアプリとは趣の異なる目立つアイコンができあがる可能性が大きくなります。



図 7-4 名前を重視したアイコンの例。左から Aardvark Mobile、RedLaser、Birdhouse、MoneyBook、Ego、Cellar Rat。

この種のアプローチは、すでにブランドが確立している組織には特に有効です。ユーザーはひと目ただけで意味がわかるので、ロゴに頼ることができます（図 7-5）。



図 7-5 ロゴそのものを用いたアイコンの例。左から Yellow Pages、FedEx、Twitterrific、myStarbucks、Flickr、NASA。

知名度が非常に高いブランドは、1文字か2文字のシンプルなロゴでもひと目でわかります。周知のとおり、アルファベットには限られた数の文字しかありません。つまり、早く文字をとらないとほかのアイコンにとられてしまいます。このアプローチは、すでに確立した文字ロゴと覚えやすい字体をもつブランドにとってはとても効率のよいものとなります。



図 7-6 ロゴを使ったアイコンの例。左から New York Times、Facebook、GQ、PayPal、Skype、Wall Street Journal。

アイコンに文字を詰め込みすぎないようにしましょう。1文字か2文字がちょうどよいのですが、有名なロゴ（FedEx、Yelp、NPRなど）であれば文字数が多くても大丈夫です。アイコンにアプリ名を用いたら、その上に単語や説明文を追加するのは避けましょう。視覚的に表現するのがよいのです。文字に頼ってしまうのは試行錯誤不足というものです。App Store や iPhone の画面でアプリ名を示す文字列がいつもアイコンと一緒に表示されることを忘れずに。アイコンの上にわざわざ名前を書く必要はないのです。いちばんの問題は文字を入れるとアイコンの見た目が細かくなってしまい、

「iPhone を検索」の画面や「設定」アプリで表示される縮小アイコンできれいに表示されないことです。画面上にあるアイコンを見るときにもアイコン内に書かれた文字はノイズに見えてしまいます。それが縮小アイコンとなると、もはや読みにくい落書きでしかありません。

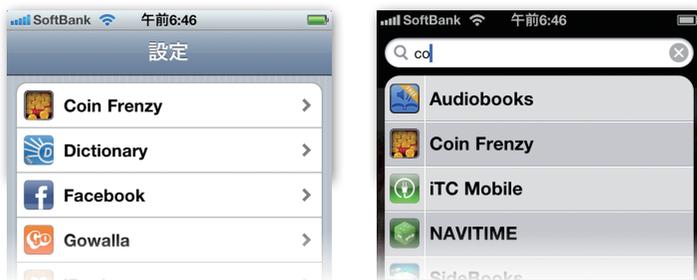


図 7-7 「設定」アプリ(左)と「iPhone を検索」(右)の画面では縮小アイコンが表示される。下にあるように、アイコン内の文字はフルサイズのアイコンでも読みにくい。「設定」や「iPhone を検索」の画面に表示されるアイコンでは文字が読めない。

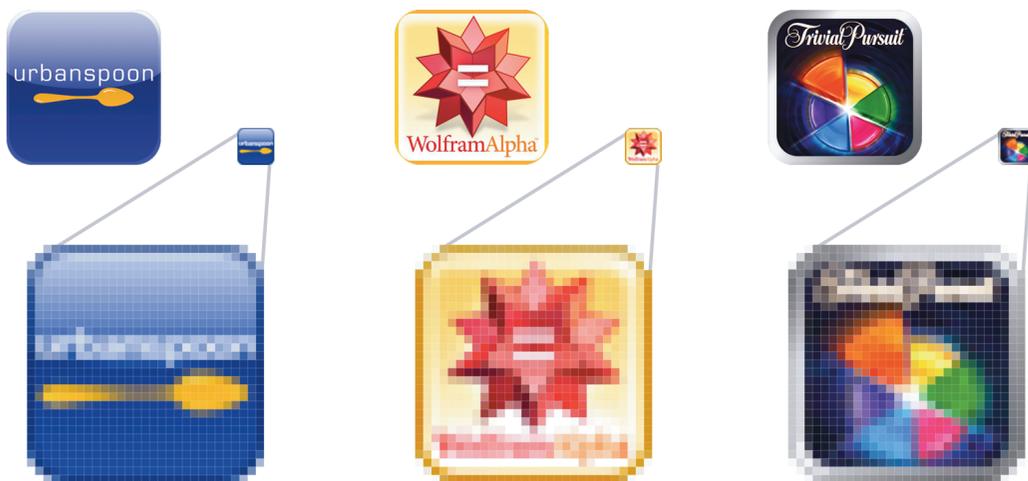


図 7-7 では、文字のせいでアイコンの輪郭がはっきりしなくなっています。よいアイコンは確固とした形をもっています。華やかなライトを使った効果やリアルな 3D などを用いた表現を使うのも（それができるなら）悪くはありません。しかし、もっとも大事なものは基本的な形の輪郭です。基本は、ひと目でそれとわかるような単純な形であるべき、ということ。自作アプリのアイコンを影絵にしたとして、それを見ただけでわかるようになっていれば、まずまず順調な出だしといえます。いくつもの形を合わせてイメージをごちゃごちゃしたものにするのはやめましょう。ふたつ以上の形を使わざるを得ないときは、それぞれが明確に判別でき、重なった部分がかっきりとわかるようにしましょう。フルサイズのアイコンは、6 章で見たツールバーアイコンよりも凝った作りにもできますが、基本ルールは同じです。つまり、意味が明快であることが重要なのです。

会社などのロゴをデザインする際には、すっきりとしたシルエットをもつようにすることが大切ですが、アイコンはいわばアプリのロゴなのです。アイコンとインターフェース、この両者のデザインでアプリの「ブランド」が決まります。複数のアプリからなるアプリ群を提供しようと考えているなら、

すべてのアプリに共通する「ブランドイメージ」を構築する必要があるでしょう。

アプリ開発企業の Tapbots と App Cubby はかなり違ったインターフェーススタイルをもつアプリを作っていますが、Tapbots ではレトロな機械風、App Cubby では木目調のデザインが使われており、それぞれのアプリのアイコンには統一感があります。また、路線探索アプリを作っている iTrans では、よく似たアイコンを使いつつ、都市の名前と窓に映る街のシルエットを変えることでアプリを区別しています。



図 7-8 左から App Cubby、Tapbots、iTrans のアイコン。いずれも統一感をもった「ブランド」をうまく表現している。

## 7.2 | 自作アプリのアイコンを作る

アプリのアイコンには、童話『三びきのくま』のように「大」「中」「小」3つのバージョンがあります。ユーザーがいちばんよく目にするのはホーム画面でタップして起動する「中」サイズのものですが、残りふたつのサイズにも重要な役割があります。まず「大」ですが、App Store ではこれをもとにしてアプリの詳細画面に表示する大きなアイコンを作っています。「小」は iPhone の「iPhone を検索」の結果や「設定」アプリで表示されるものです。さらにもう少し話を複雑にしているのが、(iPhone ではなく) iPad 用の「小」と「中」のアイコンはまた別の大きさだという点です。このため、iPhone でも iPad でも動く「ユニバーサルアプリ」を作る場合は、iPad 用にもう 2 種類の大きさも用意する必要があります。さらにさらに話を複雑にしているのが、iPhone 4 でさらに別のサイズが加わったことです。

iPhone 4 (Retina ディスプレイをもつ機種) 用も含めて表にまとめると表 7-1 のようになります。

アイコン	サイズ (ピクセル)
iPhone 用の小	29 × 29
iPhone 4 (Retina ディスプレイ) 用の小	58 × 58
iPad 用の小	50 × 50 (表示されるときは 48 × 48)
iPhone 用の中	57 × 57
iPhone 4 (Retina ディスプレイ) 用の中	114 × 114
iPad 用の中	72 × 72
大	512 × 512

表 7-1 アプリアイコンのサイズ

表 7-1 を見るとわかるようにアイコンの大きさにはずいぶん差があって、「大」は iPhone 用の「小」の 20 倍近くあります。つまり iPhone 用の 3 種類のサイズに適したディテールはそれぞれ非常に異なるということです。アイコンの図案をデザインするときはまず「大」から始め、その大きなキャンバスを生かしたディテールや明度にしましょう。「大」で目指すのは、App Store に並んだ場合に人目を引く効果的な看板となることです。ただ、iTunes にアプリが並ぶときは縮小されて表示されるため、「できあがった」と宣言する前にアイコンを 175 × 175 ぐらいに縮小しても見栄えが悪くならないかどうか確認しておきましょう。

次に「中」を作りますが、「大」をもとにして、縮小するとつぶれてしまうディテールを削ります。このときもシルエットに注意しましょう。ホーム画面で表示されるアイコンですから、ユーザーがひと目でわかるよう主要部分を見分けのつきやすい鮮明な形にする必要があります。同じことが「小」についても言えますが、「中」よりもさらに鮮明で見分けのつきやすい形にしなければなりません。29 ピクセル四方ですから中身を細かくする余地はなく、「大」をそのまま縮小したりすれば色の濁った汚いものになってしまいます。デザインにもよりますが、「小」でもシルエットがはっきりするよう、最初から作り直したほうがよいかもしれません。

こうした作業をすべて行くと、複数の微妙に異なるアイコンができることが多いのですが、だからといってやたらに異なるデザインにしてよいわけでもありません。App Store のレビューで「まぎらわしい」と批判されることがあるのです。細部が多少違って、ぱっと見て「大」「中」「小」がどれも同じに見えるようにする必要があります。ただし App Store 用の「大」のアイコンに「Sale」の札など販促のメッセージを付けることは Apple でも認めています。ちょうど店の窓に「Sale」の札をぶら下げるのと同じです（でも気をつけてください。こうした販促の仕掛けで安っぽい感じになってしまう恐れがある上、デザインも損なわれがちです。商売と品のよさは両立しにくいものですが、試していけないわけではありません。ただ、アイコンに札を付けるときは慎重に）。

アプリのアイコンは必ず角が丸くなった形で表示されますが、これは App Store や iPhone が舞台裏で自動的にそうしているためです。ですから自分でアイコンの角を丸くしたりせず、単純な正方形の画像を使ってください（画像は 24 ビットの PNG 形式で非透過にしましょう。透明ピクセルは黒



図 7-9 Wikipedia を読むためのアプリ Articles のアイコンは、「大」「中」「小」いずれも最高の出来栄を狙ってゼロからデザインされた。

に置き換えられるので、ホーム画面の壁紙をカスタマイズしている場合、あまりきれいに見えません。

アイコンには自動的に光沢がかかるようになっていますが、そうならないようにしましょう。これは Apple がすべてのアイコンに宝石のような統一感をもたせるべく善意で行っている操作で、統一感という考え方は尊重すべきものの、自分で工夫しさえすればつけられるライティング効果を、自動フィルタによってかけてもらうのはやめましょう（図 7-10 の例を参照）。Apple の標準のアプリにさえ、こうした標準の光沢処理をしていないものがあります。皆さんのアイコンにもライティング効果を施して、デザインに合ったツヤを出しましょう。自分のアプリの不可欠な要素を人任せにはしてはいけません（自動で光沢がかからないようにするには、アプリの Info.plist ファイルに `UIPrerenderedIcon` キーを追加します。詳細は『iOS アプリケーションプログラミングガイド』[<http://developer.apple.com/jp/devcenter/ios/library/documentation/iPhoneAppProgrammingGuide.pdf>] を参照してください)。



図 7-10 標準の光沢処理を施したアイコン（上）には統一感はあるが、独自のツヤ処理をしたアイコンや、光沢もツヤも一切出していないアイコン（下）のような独自性はない。

## 7.3 | 名前の役割

ワッフルの店を見つけるアプリを何と呼ぶでしょうか。Wafflr でしょうか、iWaffle でしょうか、それとも Whence the Waffle でしょうか。名前の重要性は意外に低いものです。少なくともユーザーの観点から見た場合は。もちろんマーケティング面では重要です。覚えやすい名前は口コミでは決定的です。アプリのブランド化に効果的ですし、キーワード検索でのヒットの度合いにも影響します。ぜひともマーケティング担当者にしばらくの間名前をあれこれ検討させて、できるだけのことをやらせてください。とはいえ、一旦 iPhone でアプリの居場所が定まってしまうと、ユーザーは名前よりもアイコンや機能で認識するようになります。たとえば iAwesome という文字ではなく、派手なオレンジ色のアイコンを探すのです。ユーザーの立場から見ると、アイコンとインタフェースのほうがどんな名前よりもはるかに重要です。画面上では名前はアイコンという「図」に添えてある「キャプション（説明の文章）」にすぎません。

この意味では**名前は短くする**のがベストでしょう。表示可能なスペースに名前が入りきれないと、iPhone は名前の真ん中の部分を削って詰め込んでしまいます。Local Concerts は Local...ncerts になり、Whence the Waffle は Whenc...affle になってしまうのです。このように、無害で便利なラベルだった名前が混乱の元になってしまうわけです。App Store ではホーム画面用に省略名を使うことも認められています。つまり、iTunes では目立つようにフルネームを、ホーム画面用には省略名を登録することができます。たとえば ToDo リストアプリ Remember the Milk は RTM、Dragon Dictation は Dictation、宅配便追跡アプリ Delivery Status Touch は Deliveries と表示されます。必要なことはなんでもやって、名前が真ん中で削られることのないようにしましょう。

ホーム画面で使われているフォントの幅はどれも同じわけではないので、何文字の名前なら入りきると断言はできませんが、Safari を使えばアプリを「ビルド」しなくても長さの確認ができます。Safari でならウェブページをホーム画面のアイコンとして登録できますから、好きなウェブページへ行き、ツールバー中央にある「矢印ボタン」をタップして [ホーム画面に追加] を選択してください。試したい名前をタイプすると、ホーム画面にアイコンができ、その下に名前が付きます。アプリのアイコンの下に名前が表示されるのと同じです。名前がうまく収まればしめたもの。そうでなければやり直しましょう。

## 7.4 | 起動画面

さて、皆さんの作ったアプリをダウンロードしてくれるという人も見つかり、思わずタップしたくなるほど魅力的なアイコンもできあがり、そのユーザーがタップしてアプリを起動したとしましょう。そこで、ユーザーはしばらく待ちます。iPhone のどのソフトウェアについても言えることですが、皆さんのアプリもロードして動き出すまでに 1 秒から 3 秒ほどかかります。決して「永遠」ではないのですが、移動中に使われることの多い iPhone の慌ただしい環境ではあたかも永遠のように

感じられかねません。どれほど長くかかろうと、本当に重要なのは**どれほど速いと感じるか**です。そうした印象は、アプリが起動するまでの間に表示する画像を工夫することで、かなり改善されます。その画像をマジシャンの巧妙なトリックのようにタイミングよく見せれば、起動時間を実際よりも短く感じさせることができます。すばらしいインタフェースデザインを駆使して時間をねじ曲げてしまう、『スター・ウォーズ』のジェダイも真っ青の心理操作については 10 章で詳しく説明するとして、ここではアプリの起動画面の効果について考えましょう。

アイコンをタップすると、iOS はアプリのコードを実行し始める前に画像を表示します。これが起動画像、つまりアプリがロードしている間、ユーザーに見せる画像です（320 × 480 ピクセル<sup>†</sup>の PNG 形式で、アプリのコードにバンドルしておくとも iPhone が自動的に表示してくれます。皆さんの並外れた Photoshop の腕前やたぐいまれに美しい会社のロゴなどでユーザーを幻惑し魅了するといった具合に、この画像を華々しいスプラッシュ画像にしたいという誘惑に駆られるでしょうが、そういう衝動は抑えて逆のことをしましょう。つまり、さえない画面にするのです。

起動画像はアプリがすでに起動して動いているという印象を与えるものにしてください。Apple も、起動画像はアプリの初期画面そっくりにし、コンテンツを空にする、つまりインタフェースデザインの殻だけにしよう勧めています。この方法なら、すぐにインタフェースが見えるので、コードがまだ動き始めてもいないのに、アプリがすでに起動して一所懸命コンテンツをロードしていると思わせることができます。アプリが本当にコンテンツをロードし終わると、起動画像に代わって本当の初期画面を表示します。あたかも最初に表示された空の画面を実際のコンテンツで埋めたかのように。

この「にせの初期画面」戦略を、イラストやロゴをあしらった豪華なスプラッシュ画像を起動画像として使う方法と比べてみてください。後者はまるで広告です。にせのインタフェース画像が「私はあなたのために働いていますよ」というメッセージを伝えるのに対して、ロゴのほうは「まず私の自己紹介をさせてください」と主張しています。本質的な作業内容は同一なので、どちらの画像のロードにも同じ時間がかかるのですが、心理的な違いは明白です。味気ないインタフェース画像のほうが起動時間が短いように**感じられる**のです。それにユーザーの仕事に対する敬意も表すことにもなりません。マーケティング的に見てメッセージや画像がどれほどすばらしいものであろうと。

むしろ「すばらしければすばらしいほど」問題が大きいとさえ言えます。起動画像がユーザーの注意を引けば引くほど、起動時間の長さに注意が向き、意図的に起動を長びかせているのではないかと思わせてさしてしまうのです。まるで、1999 年頃に作られた、最初に Flash ムービーで自社のロゴを見せびらかすウェブサイトであるかのように思われてしまうのです。さらに「完璧なもの」にしようと言わんばかりに、起動画像の直後にアニメーションを表示するアプリすらあります。PayPal というアプリでは、操作ができるようになるまでの 1 ~ 2 秒間はコインが弾むアニメーションを流していましたし、(最初のアニメがなければすばらしい) Virgin Atlantic の Flying Without Fear (飛行機恐怖症を克服しよう) というアプリでは、飛行機恐怖症を克服するためのリラックスエクササイズが

<sup>†</sup> iPhone 4 など、Retina ディスプレイをもつ機種では 640 × 960 ピクセル。



図 7-11 起動画像はアプリの初期画面のコンテンツを空にしたものにするるとよい。標準の「天気」アプリ（上）の起動画像もこの原則に従っており枠だけが表示される。「設定」アプリ（下）でも、空のナビゲーションバーと細い縦縞の背景が表示される。

始まる前に、アニメーションを数秒間流します(時速 1,200 キロメートルで飛ぶ飛行機の通路を転がっていく自分を思い浮かべて過呼吸に陥っているときには著作権表示のアニメーションなど見たくもないでしょう)。

こうしたアプリは、実は準備ができるまで舞台裏で重労働をしていて、段取りがつくまでアニメーションで時間を稼いでいるのかもしれませんが、10 章で解説してありますが、こういう場面では単純なプログレスバーなど準備の進行状況の表示のほうが適役でしょう。残念ながら、スプラッシュ画像を表示すると、アプリが自己宣伝のために時間稼ぎをしているように見えてしまうのです。おまけに、この種の視覚的な効果には、ユーザーがアプリを「起動中である」ことを強調してしまうきらいがあります。たしかに実際にも起動中ではあるのですが、むしろここでは、すでに起動しているアプリに「切



図 7-12 PayPal も Flying Without Fear も起動までの数秒間アニメーションを流すが、そのために起動時間がかえって長く感じられてしまう。

り替えている」とところで、切り替わるのを待っているという印象を作り出すべきなのです。このトリックについては次の節で解説します。

## 7.5 | スリープ機能

今重い病気に苦しんでいる人なら、自分が「冷凍人間」になって何年も経ってから解凍してもらって病気を治してもらいたいと思いませんか。ウォルト・ディズニーの遺体がディズニーランドの地下の一室で蘇生する日を待っているという有名な都市伝説もあるようです。しかし、(残念ながら)現在の技術では、冷凍人間を春の花のようにみずみずしく、すぐにでも動き出せる状態で生き返らせることはできません(ちなみにウォルト・ディズニーは 1966 年に火葬されてしまいました)。しかし、iPhone の世界では独自の 방법으로、似たようなことができるようになりました。iOS 4.0 から、アプリを終了する際に「コールドスリープ冷凍保存」してくれるようになったのです。つまり、アプリをもう一度動かすと、前回終了したときとまったく同じ画面に戻れるようになったのです。

実は、この機能が導入される前から、アプリを行き来する際にはこの冷凍・解凍が行われているのだと多くのユーザーが思っていました。パソコン用のソフトウェアでは起動や終了に数秒程度の時間がかかることが多いのですが、iPhone のアプリは起動も終了も素早いので、どのアプリもあらかじめロードされていて、すぐ使える状態なのかもしれないという印象を与えていました。とても終了したり再起動したりしているようには思えなかったのです(もちろん実際にはそうになっていたのですが)。現在は冷凍・解凍が可能になっていますが、アプリを「冷凍保存」しておけるケースは限られています。iPhone や iPod のすべてのハードウェアでアプリをサスペンドしておくだけのパワーがあるわけではなく、またそのパワーがある場合でも一度にサスペンドできるアプリの数は限られています。いずれの場合も、あるアプリに戻るといことは、もう一度起動することにほかなりません。もっ

とも、ユーザーはこんな技術的な細かい差など問題にしません。内部で何が起きていようと、前回やめたところからアプリが再開してくれればそれでよいのです。アプリの起動時には(たとえ本当は違っても)「冷凍保存された状態」から復帰しているんだという印象を与えるよう、ひたすら努力する必要があります。

このことが iPhone のコードを書く開発者にとって何を意味するかというと、アプリがうまく起動できるかどうかは、うまく終了できるかどうかにかかっているということです。アプリが終了するときには、保存されていないデータだけでなく、インタフェースの状態に関する情報、つまり現在表示しているビュー、位置、コンテンツなども大急ぎで保存する必要があります。そういった情報をしっかりと保存しておけば、再起動したとき今まで背後でずっと動いていたかのようにインタフェースを正確に元の状態に戻せるのです。

こうしたトリックの鍵となるのが、ここまで見てきたように、巧みにデザインされた起動画像です。しかし画面のデザインが数種類あるアプリでは、起動画像がいつでも役に立つとは限りません。起動画像はひとつしか登録できず、起動時に交換することもできません。起動画像をインタフェースに見せかけることができるのは、すべてのビューで一貫した背景をもつアプリの場合だけです。画面の背景やレイアウトが複数あってそれぞれが互いに大きく異なるアプリでは、復帰したときに起動画像とその時点のレイアウトが一致しない可能性があるため、見た目が食い違って起動がおかしな具合になります。たとえば科学計算機アプリ PCalc (3章の開発者は語る「James Thomson と PCalc」)ではキーボードのレイアウトが数種類あるため、起動画像を表示する際に違うレイアウトを表示してしまうリスクが常につきまといます。開発者の James Thomson 氏は起動画像としてアイコンの画像を表示するように戻しました。「これで食い違いはなくなりましたが、ユーザーはスプラッシュ画像をじっと見つめながら計算機が現れてタイプできるようになるのを待つ4秒間を永遠のように感じていました。それどころか、スプラッシュ画像をこんなに長く表示するなんて、なんというぬぼれ屋だと非難するメールまで来たんです」。

Tomson 氏は賢いハイブリッドな解決法を編み出しました。PCalc が終了する前にその時点の画面のスナップショットを撮り、画像として保存するようにしたのです。再び起動されると、起動画像(ロゴのスプラッシュ画像)を短い間表示し、その間に前回保存されたスナップショットを取得します。そしてこの画像を表示してから、ようやく PCalc は起動コードを動かして自分自身の準備を始め、それが完了すると表示していた画像を「本物の」画面と交換します。この手品のおかげで、前回終了したところからほとんど瞬時に起動するという錯覚を演出しているのです(すばらしいことに PCalc はこの「にせの画像」を表示している間にタップされた内容まで記憶していて、コードが完全にロードされるまでは何も起きないものの、ロード完了後にはタップされた数字が表示されるようになっていきます)。PCalc がロゴ画像を一切表示しなければまさに完璧なのですが、残念ながら起動画像を交換することができないので、インタフェースのスナップショット画像が表示できるようになるまでは、いつも同じ PCalc のロゴを表示せざるを得ません。



図 7-13 PCalc には画面が数種類あるため起動画像に「にせのインタフェース」を使うことができず、ロゴを表示する(左)。この画像はほんの短時間表示されるだけで、前回の終了時のスクリーンショットに置き換えられる(中央)。なお、処理速度の遅い機種ではアプリのロードが終わるまでの間、冗談めかしたメッセージが表示される(右)。

## 7.6 | 初期画面

皆さんのアプリのお客様はアイコンをタップし起動画像も目にして、やっと最初の画面にたどり着きます。さて、それで？皆さんのアプリがあまりにも使いやすく直感的なので、どうやって使うかが誰にでもひと目でわかってしまう、というのが理想でしょう。単純なアプリや、現実の世界から借りてきたイメージを利用した「超」わかりやすいインタフェース(たとえば、Air Hockey アプリ)なら、それほど悩むこともありません。しかし高度な機能が提供されているアプリや複雑な作業ができるアプリの場合は、たとえインタフェースのデザインが非常にすぐれていても、若干のヒントがありがたいものです。特に皆さんのアプリが初めてユーザーの目に触れるときにはそうしたヒントが有用です。初めてのデートでは雰囲気作りが肝心ですから、皆さんのアプリも礼儀正しくユーザーを歓迎しなければなりません。初期画面はユーザーを歓迎する責任を負っているのです、初めて表示するときには少し変えてあげると効果的です。この特別な初期画面は「玄関マット」だと思ってください。ユーザーが初めて目にする皆さんのアプリの画面を、より魅力的で有益なものにするために重ね敷きする玄関マットです。

これは最初の画面が空白で始まるような場合に特に重要です。空の画面などが出てきたらユーザーはまごつくでしょうから、初回に一度だけ、データを追加していく手順をユーザーに教えてあげると便利でしょう。たとえば Wikipedia を読むためのアプリ Articles は記事を検索するまでは画面に表示するものが何もありません。同様に、チェーン店のメンバーズカードを管理するアプリ CardStar では、最初のカードを登録するまでは見るものが何もありません。どちらのアプリでも最初の画面は空白で



図 7-14 Articles (左) と CardStar (右) の最初の画面は初回起動時には空白になるが、作業の始め方についての簡単な説明が追加される。

すが、作業の始め方についての簡単なメモを「玄関マット」として追加してあります。皆さんのアプリでも最初の画面にコンテンツを追加するための動作が必要なら、玄関マットを使ってどこから始めるべきかのヒントを示しましょう。

オンラインアカウントが必要なアプリでも、初対面のときには少しでも特別な礼儀作法が必要です。初めてのユーザーが到着したら、アカウント情報の入力方法がわからず設定画面を探しに行く、といった事態に追い込んではいけません。初めての訪問者用の画面を表示し、ユーザーに自己紹介を促しましょう。サインインしてもらったり新しいアカウント情報を登録してもらったりするのです。繰り返しますが、ユーザーにとってはこの画面こそが皆さんのアプリの第一印象となるのですから、魅力的でわかりやすく親しみやすいものにしましょう。ユーザーに初めてアカウントを作ってもらう場で

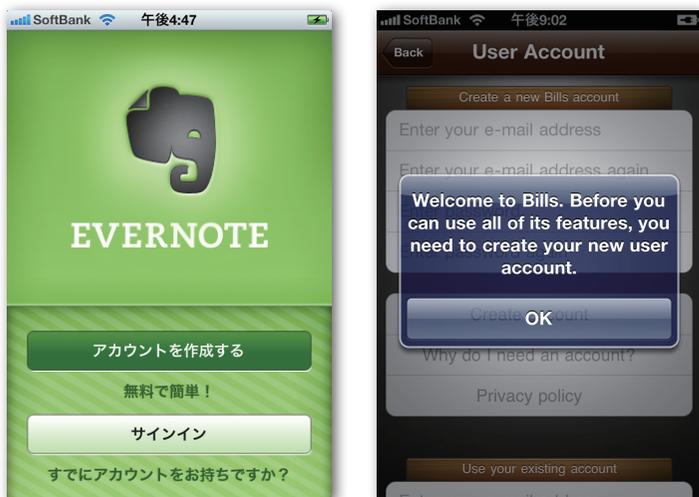


図 7-15 メモ用アプリ Evernote (左) では、初めてのユーザーにサインインや登録を促す画面を用意している。Bills on Your Table (右) は「まずまずの線」を行っているのだが、アラートボックスで状況説明をしてから設定画面に連れて行かれる。専用のウェルカム画面を作るほうがよい。

はアラートボックス（「10.1 割り込みのタイミング」）や素っ気なくみずばらしいレイアウトは避けなければなりません。

## 7.7 | マニュアルは万能ではない

1980年代アメリカで、ほかのみんながテレビドラマ『ダラス』に釘付けになっているとき、私が好んで観ていたのがTVシリーズ『UFO時代のときめき飛行—アメリカン・ヒーロー』です。不器用な学校教師ラルフ・ヒンクリーがエイリアンたちと鉢合わせし、世界を救うようにと特殊能力のあるスーパースーツを与えられます。ところが思いがけない問題が。マニュアル（取扱説明書）を落としてしまうのです。そのため手探りでスーパーマン稼業を始め、毎回スーパースーツの超人的な機能を試そうとするのですが、空を飛ばそうとしているのに宙返りをしてしまったり、予期せず新しい機能を働かせてしまったりします。見ていてもどかしいながらも大笑い。ラルフに使い方がわかりさえすれば、そのスーツにできないことなどないと視聴者は知っていました。奇しくもパソコンの黎明期で、この番組の最初のシーズンの真っ最中にIBM PCが発売され、このドラマは科学技術の驚くべき可能性とそれを使いこなせない人間の苛立ちを予見する寓話となったのです。

この主人公がマニュアルさえなくさなければ！問題はここです。すぐれたマニュアルは必ずしも（いや、むしろたいていは）答えにならない。エイリアンのご機嫌を損ねるつもりはないのですが、スーパースーツの使い勝手に問題があることは明らかで、この点にはアプリをデザインするときにも気をつけなければなりません。「タップする価値のあるアプリ」もユーザーにスーパーパワーをもたらし、スケールの大小こそあれ「スーパーな」存在になる手助けをしてくれます。使いやすいアプリなら特別な配慮をしなくても苦もなく自然に日常に溶け込み、スーパーヒーローの誕生となります。ユーザーが皆さんのアプリと初デートをするときはすべてがスムーズに運ばなければなりません（ラルフが初めて飛んだときには壁に激突してしまいました）。アプリにとって主となる仕事は、少し試したり、初回に表示する玄関マットのヒントを見たりするだけで使えるよう、簡単でわかりやすいものにする必要があります。手早く使うモバイルアプリに不可欠なのは使い勝手のよさなのです。

基本的な使い方を説明するヘルプ画面を追加しようと考えている場合、まずはデザインを再検討して、主要な機能へのアクセス方法の改良、画面の整理、重要なタスクのハイライト、ラベルの明確化



といったことを行いましょう。メインの機能を使おうとしていつもつまづくのであれば、何かがおかしいのです。そのレベルだと詳しい説明を画面に載せても役に立たないでしょう。これは全世界のテクニカルライターにとって失望の種となってしまふことなのですが、ほとんどの人がソフトウェアの説明書を読みも求めもしません。このことはモバイルアプリには特によく当てはまり、ユーザーは考えたり余計なものを読んだりせずに済む、素早く、そしてスムーズな使い心地を期待しています。

だからといってヘルプ画面を用意すべきではないということではありません。使い始めるときに**絶対読まなければならない**という風にはすべきでないと言っているだけです。より高度なアプリであったり微妙な使い方をしたりする場合には、高度な機能についての追加説明が必要かもしれません。使い方のコツやヒントがあれば、初心者が使い慣れるのに役立ちますし、「設定」画面にヘルプを追加したり、モバイルサイトへのリンクを載せて、そこでチュートリアルを見てパワーユーザーになってもらったりするのもよいでしょう。しかしこの方法の欠点は、説明書が「アプリの部屋のほこりっぽい片隅」に追いやられてしまうためユーザーはよほどのことがないと探してくれないということです。普通のユーザーや初心者が、こういう説明書を「偶然」見つけるなどと期待してはなりません。

もっと便利な方法は、インタフェース上のよく使われる場所にヘルプを組み込んでしまうということです。たとえばメモ帳アプリ Evernote では新規メモの画面に Tips タブを付けて、よりクリエイティブな使い方を提案しています。これは時々ユーザーに試してもらって「へー、これいい！」と言ってもらうための助言をするには理想的な方法です。もっとも、わずか一度しか見る必要のない初心者用のヘルプには適していません。ユーザーがめったに使わないような要素については、それを主要な画面に詰め込んでではありません。



図 7-16 Evernote の新規メモの画面 (左) の右下には Tips のタブがある。ここをタップすると珍しい使い方や高度な機能を詳しく説明する画面が表示される (右)。

時として避けられないこともあります。アプリによっては簡単な紹介メモ以上のものが必要になる場合があるのです。あまり知られていない概念を導入しているアプリでは、ユーザーの手をとって導

いてあげるとよいでしょう。つまり玄関マットをさらにいいものにするとする方法が効果的なのです。楽しい iPhone 用ゲーム Backwords は説明するのが難しいのですが、たった一度でもやってみれば簡単にプレイできるようになります。大筋を紹介しましょう。第 1 のプレイヤーが部屋の外に出ているうちに、第 2 のプレイヤーが秘密のフレーズを録音します。第 1 のプレイヤーを部屋に呼び戻し、録音した言葉を逆転再生して聞かせますが、変な音になっていて何と言っているのかわかりません。第 1 のプレイヤーはそのめっちゃくちゃな音を真似して、マイクに向かって唸ったりペチャクチャやったりしなければなりません（これがとてもおもしろいのです）。最後に 2 番目の録音を逆転再生しますが、うまく真似できていれば、だいたい最初に録音したフレーズのように聞こえます。こうして最初のフレーズが当てられれば点がもらえるというのがルールです。

ふう。これは文字どおり「百聞は一見にしかず」のゲームで、開発者の Shadi Muklashy 氏はこう言っています。「開発中は週末が来るたびにみんなで Backwords パーティをやっていましたが、自分の番になって説明書を読もうとする人なんてひとりもいませんでした。部屋を見回してこう尋ねるんです。『よし、それでどうすればいいんだい?』と」。Muklashy 氏はヘルプ画面として、それぞれのステップに解説用のレイヤーをつけて上に重ねて表示しました。解説はくだけた感じの大きな手書きのフォントで書いてあるので雰囲気はなかなかよく、ゲームショーの司会者がそれぞれのステージを説明してくれているような感じです。そしてゲームのコツがつかめたら、設定で解説をオフにします。



図 7-17 Backwords には解説を読みながらプレイするインターフェイスが内蔵されている。まずは逆転されて流されるフレーズを聞きながら元のフレーズが何であるかを考え（左と中央）、その後、逆転されたフレーズを真似して録音するが（右）、その際タップすべき場所とその理由を示す手書きの解説が出る。

## 7.8 | トップ画面

すっきりしたレイアウトとわかりやすい言葉づかいはアプリのどの画面でも必要不可欠ですが、トップ画面では特に大切です。ユーザーは初めて使うときもその後もトップ画面を頼りにして自分の位置を把握し、おおよその使い方や機能を知るものです。この画面の仕様は、選択したナビゲーションモデル（「4.2 画面遷移 — Appleのナビゲーションモデル」）やインタフェースメタファー（「6.6 メタファーで考える」）によって決まりますが、次のような一般原則は大いに役立つでしょう。

### ●技術者向けの専門用語を避ける

わかりやすい「キャッチコピー」を考えましょう。開発者の視点ではなくユーザーのニーズに焦点を合わせましょう。わかりにくい専門用語を避け、ユーザーになじみのある言葉づかいでわかりやすい文章を書くようにしましょう（図 7-18）。

### ●ダッシュボードを検討する

データを集めたり表にしたりするようなアプリでは、最初の画面で集計表やシステムがひと目でわかるような「スナップショット」を用意する方法を検討してみましょう。この手法は、たとえば会計アプリ、フィットネス日誌、カロリー記録など、いろいろな個人データを集めるアプリで非常に有用です。このようなアプリではばらばらなデータ項目を多数扱うので、情報を一覧できるダッシュボード画面にすれば、個々のデータではなく、その全体をまとまった形で見せることができます。口座にお金がいくらあるか、カロリーをどれぐらい消費したか、といった具合に（図 7-19）。

### ●視認性テストを行う

スクロールなしのユーティリティ・アプリの出来栄えを測定する視認性テスト（「3.8 スクロールを疑え」）を覚えていますか。大半のアプリの初期画面についても同様の原則が適用できます。

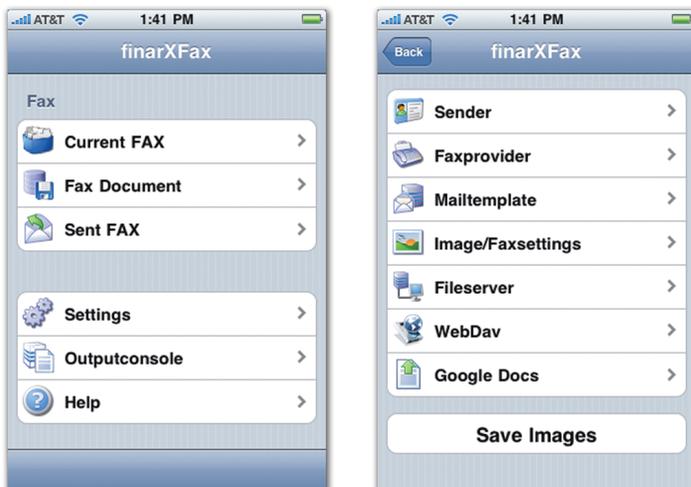


図 7-18 finarX Fax は iPhone からファックスを送れる有用なアプリだが、技術用語が玉にキズだ。「Outputconsole（出力コンソール）」ではなく「Fax History（送信履歴）」に、「WebDAV」の代わりに「Web Disk」や「Online Account」を使うといった具合に、わかりやすい言い回しにすれば親しみやすくなる。



図 7-19 MoneyBook(左)と Mint.com Personal Finance(右)では、初期画面に家計の状況がひと目でわかるダッシュボードが用意されている。

トップ画面は、そのアプリでどんな機能が利用できるのかがひと目でわかるような構造になっていなければなりません。アイコンをうまく使うと表形式のリストを整理するのに役立ちますし、特に機能が多くのアプリではインタフェースに写実的なメタファーを使うとわかりやすくなります。たとえば Facebook では iPhone のホーム画面のレイアウトスタイル（スプリングボード）を真似て、機能やコンテンツのカテゴリをアイコンにし、それをグリッド表示しています。このレイアウトは見栄えがすると同時に全体を見渡しやすいものでもある上、このアプリがアプリ風機能の集まりだということを強調してもいます。章末の「開発者は語る」に、Facebook アプリの開発者である Joe Hewitt 氏が語ってくれたアプリのデザインと開発秘話を掲載しました。



図 7-20 Facebook 3.0 (左)では iPhone のホーム画面風のナビゲーションをトップ画面に採用した。このようにアイコンを並べる方法は、その後 LinkedIn (中央)や Yelp (右)などでも採用された。

## 7.9 | 押さえどころ

- ✓ アプリのアイコンは機能、インターフェース、名前、ブランドを表すものにしよう。
- ✓ アイコンのデザインには力強く単純なシルエットを取り入れよう。
- ✓ 短い名前がベスト。
- ✓ 素早く起動するかのように見せるため、起動画像をアプリのバックグラウンド画像に似せてカモフラージュしよう。
- ✓ 「一時停止したアニメーション」による錯覚を利用しよう。
- ✓ 初めてのユーザーのために単純でわかりやすい「玄関マット」を用意しよう。
- ✓ アプリの初期画面は特にすっきりとしたものにしよう。できればダッシュボードを使おう。

開発者は語る

## Joe Hewitt と Facebook

世界一人気のあるソーシャルネットワークサービスは、世界一人気のある iPhone アプリでもあります。Facebook は App Store の初年度に最多ダウンロード数を記録したアプリです。元のウェブサイトの膨大なユーザー基盤と肥大化した機能群が、開発者の Joe Hewitt 氏にとっては大きな難題となりました。iPhone 用にバージョン 3.0 までの Facebook を開発したときの開発秘話は、まさしく Hewitt 氏の「独り舞台」です。大半の iPhone アプリは機能セットが小ぶりですが、Facebook のユーザーはウェブサイト版の全機能が使えることを望みました。最初のバージョンは機能が限られていましたが、Facebook 2.0 から 3.0 にかけては使える機能が急速に増えて、ウェブサイト版の機能をほぼすべて使えるようになりました。Hewitt 氏は大きな機能セットを小さな画面に押し込めるといった難題とその解決策について次のように語ってくれました。



### ライト版以上のもの

iPhone を初めて手にしてから約 6 ヶ月後、もうパソコンは必要ないと思うようになりました。パソコンでできることはほぼ何でも iPhone でできるだろうと。小さな画面のほうがよいことはたくさんあります。デザイナーが本当に重要なことに焦点を絞らざるを得なくなります。私は考えが変わりました。最初 iPhone は「コンパニオンデバイス」だと思っていたのですが、新しいプロジェクトや会社を始めるときに使うべきデバイスだと思うようになったのです。つまり、まず iPhone 向けに開発して次にパソコンのウェブ向けに開発するというわけです。そこが我々の向かう場所だと思います。いつも持ち歩いている携帯タッチスクリーンデバイスでのほうがうまくいくことが山ほどあります。iPhone アプリを「ライト版」だと考えるべきではないのです。

Facebook アプリの最初のバージョンではもっとも重要な機能を作る時間しかありませんでした。App Store が始まったときで、ユーザーの期待がどんなものか誰にもわかりませんでした。やがてその期待が非常に高いことがわかりました。限定された機能しかないウェブサイト版のコンパニオンアプリでユーザーが満足するものとばかり思っていたのですが、私の完全な見当違いでした。Facebook に関してはパソコンで使える機能が全部使えることを期待していたのです。今では私もちらかといえば同意見ですが。

## 「サブアプリ」の集合

iPhone アプリは一芸に秀でた存在であるべきだという一般的な考え方には間違いなく賛成ですが、このことは概して Facebook には当てはまりません。ユーザーが求めていたのはウェブサイトの機能が全部使えるアプリでした。こうした要望とシンプルさを両立させるため、私はアプリをサブアプリに分割し、さらに個々のサブアプリの機能をできるだけ絞り込んでいきました。ここから Facebook 3.0 のホーム画面におけるアイコンのグリッド表示が生まれてきました。



図 7-21 iPhone のホーム画面で使われているグリッド表示にヒントを得て Facebook 3.0 用に作成したトップ画面のドラフト 3 種。全体のコンセプトがほぼ固まったのが左で、中央のデザインを経て、最終デザイン（右）となった。

Facebook はそれ自体がプラットフォームです。私は気がついたのです。「写真」「連絡先」「メモ」など iPhone に組み込まれている主要なアプリが、Facebook にも組み込まれているんです。そのことを考えれば考えるほど Facebook は iPhone そのものの「パラレルワールド」なのだと思えてきて、それをユーザーに理解してもらうには同じホーム画面を採用すればよいと思い至りました。当時としては少々冒険でしたし、気に入ってもらえるかどうかわかりませんでした。でも思いのほか評判がよくて、これはちょっとした驚きでした。みんながすでに iPhone のホーム画面で慣れていたので、新しく覚えることがまったくなかったんです。

9 個のボタンをグリッド表示することはもちろんそれほどたいへんではありませんでしたが、ホーム画面の編集機能を追加するのは多少厄介でした。iPhone のホーム画面と同じで、Facebook でもアイコンを好きなように並べ替えることができます。アイコンをタップしたままホールドするとアイコンが揺れはじめるので、好きなところに移動できます。Apple がホーム画面をそのように実装したので、自分のアプリでも安心して同じことができました。しかしこの作業を進めるうちに、Apple のデザイナーは何割のユーザーがホーム画面の編集方法に気づいてくれると思っているのだろうという疑

問がわいてきました。アイコンを押し続けるとホーム画面の編集ができるようになることに気づいていない iPhone ユーザーはたくさんいるのではないのでしょうか。

## 標準アプリのような使い勝手

Facebook は用途が非常に広い、いわば「ユビキタスなアプリ」ですから、常軌を逸したスタイルは必要ないと思っています。ほかの多くのアプリだったらかまわないかもしれませんが、手間をかけて「車輪の再発明」をするのはやめたほうがよいと思ったのです。「Apple が作った」と思わせるほどではなくても、Apple が標準のアプリで確立したのと同じスタイル、同じガイドラインに従っているとユーザーに感じさせることが重要でした。

そのように作り込む上でいちばん努力を要したのは写真ビューアだったと思います。iPhone 標準の「写真」は実にすばらしいアプリなのですが、SDK (iPhone アプリの開発キット) でそのコードが使えないと知ってがっかりしました。そこで自分で作らざるを得なかったのですが、今でもまだ Apple のアプリと同じ動作をうまく再現できたとは思えません。写真ビューアを Apple のものと同じように動作させるため何度も実験を重ねなければなりません。たとえば最大限にズームインすると限界をほんの少し超え、指を離すと限界に戻ります。これを真似るのにたいへんな時間と労力がかかりましたが、きちんと再現するにはこういった細部が本当に重要なのです。

タッチスクリーンのおかげで、実際のものに触れているような「錯覚」をユーザーに与えることができるようになりましたが、本物のように揺れたり飛び跳ねたりしてくれないと、今言ったような錯覚が台なしになってしまいます。コンピュータを使っていたのだったという現実に戻されて、現実の世界の物体を指であちこち動かしているようには感じられなくなってしまうのです。

タッチスクリーンの出現でインタフェースのデザイン方法がガラリと変わりました。パソコンでは物体が多少デジタル化された感じでも、どのみちすべてが抽象的な世界のことなので問題ありません。たとえばマウスは、人工的な機械であるパソコンを操作するためのロボットアームのようなものです。ユーザーとインタフェースの間に距離があるのです。しかしロボットアームがなくなって直接画面に触れるようになると、逼真性を増さなければなりません。そうなるようデザインしなければならないのです。

## 飾りはほどほどに

ニュースフィード画面ではいろいろな操作を可能にしなければならなかったのが本当にたいへんでした。しかもアプリの起動直後に表示されるものですから、この画面を逃すとユーザーがさまざまな機能をきちんと試してみてくれないのではないかと心配がありました。そこで盛りだくさんに詰め込んでインタフェースをゴチャゴチャにすることなくトップの機能を使えるようにしたいと思いました。この画面についてはいくつものバージョンを作って試しました。間違いなく最大の難問でした。



図 7-22 Facebook 3.0 のニュースフィード画面の変遷(左から右へ)。初期バージョンではニュースフィードのフィルタボタンがあった。最終デザイン(右)では、ナビゲーションバーの右側のボタン(Live Feed)にフィルタを組み込んで、写真などのUIパーツを画面上部のツールバーに集約できるようになった。

ニュース画面では異なるサブリストごとのアップデートをフィルタリングできるので、そこにその機能押し込む方法を考えなければなりません。第2世代では、フィルタがはるかに目立っています。フィルタはそんなに多くは使われていないだろうと思ったので、ナビゲーションバーのボタンの後ろにしまい込むことにしました。その結果、画面下のツールバーをなくすことができ、自分のステータスの更新と写真のアップロードのためのツールバーを上部に置くことができました。特定の機能をもつボタンがずらりと並んでいるアプリや、コンテンツがひどく貧弱なアプリをよく見かけます。UIパーツは上部に一つ、下部に一つまで、それ以上並べるべきではありません。それ以上になると飾りが多すぎる印象を与えるのです。そのせいで使いたくなくなるアプリが多すぎます。

## 通知機能の問題

iPhoneの通知機能(ノーティフィケーション)はいただけません。特にアラートボックス(警告用のメッセージの表示方法)は耐えられません(「10.1 割り込みのタイミング」で詳しく説明しています)。あるアプリを使っているとしましょう。今まさに文字を入力しているところです。すると、突然テキストメッセージが来たことを知らせる通知が画面にポップアップします。文字を入力しようとしていたので、間違っって小さなアラートをタップしてしまうんです。さらによくはないのは、こうしたアラートに対して即座に対応しなければならない点です。ざっと目を通しておいて、それまでやっていた作業に戻る、ということができません。メッセージが2通来た場合には、それをまとめて「メッセージが2件ある」と表示しますが、その2通の内容を簡単に読むことができません。メッセージにざっと目を通すだけということができないのです。それまでやっていたことをすっかりやめさせて

しまって、「俺に注目しろ」と要求するわけです。アプリにプッシュ型の通知をさせなければ、こうするしかないのです。

しかしアプリの内部でやるならもっとうまくできます。Facebook 内部で通知が来た場合には、画面の下部に現れ、上にスライドします。実は、これは Palm の webOS の通知機能をもとにして作りました。webOS では実にうまく実現されています。何か重要なメッセージが来ると、その内容が画面の下部に細く表示されます。タップしてそれを処理することもできますし、スワイプして消すこともできます。これを Facebook でも再現しようとした。こんな具合にちょっと余分な作業をしなければならないのですが、こういうのは嫌いではないですね。うまく解決できると楽しいのです。

