

はじめに

本書は、iPhoneアプリを作って楽しむ本です。
いろいろなiPhoneアプリを、自分の手で作っていきます。

ものを作るって、ワクワクしますよね。
でも、ちょっと不安な気持ちもします。
どうやって作ればいいんだろうとか、本当に作れるんだろう
かとか。
そういうときに大切なことは、
「まず、小さいものでいいから、作って完成させること」
です。
「ものを完成させたときの感動」というのは、想像以上に
作った本人に、「夢」と「快感」と「パワー」を与えてくれます。
今ではすごいアプリやプログラムを作っている人でも「最初
に作って動いたときの、あの感動は今でも忘れられないよ」
という人はいくらでもいます。

作っている途中で「だいたいわかったぞ、この調子で作れば
いつか完成するだろう」なんて完成する前にやめちゃうのは
ダメですよ。頭で理解したつもりになっていても、心がぜん
ぜん満足していません。投げ出すクセがついてしまいます。

なので本書では、完成できるぐらいの小さいアプリをいく
つも作っていきます。
いろいろなアプリを作って、いろいろな楽しさを体験して
いってほしいのです。

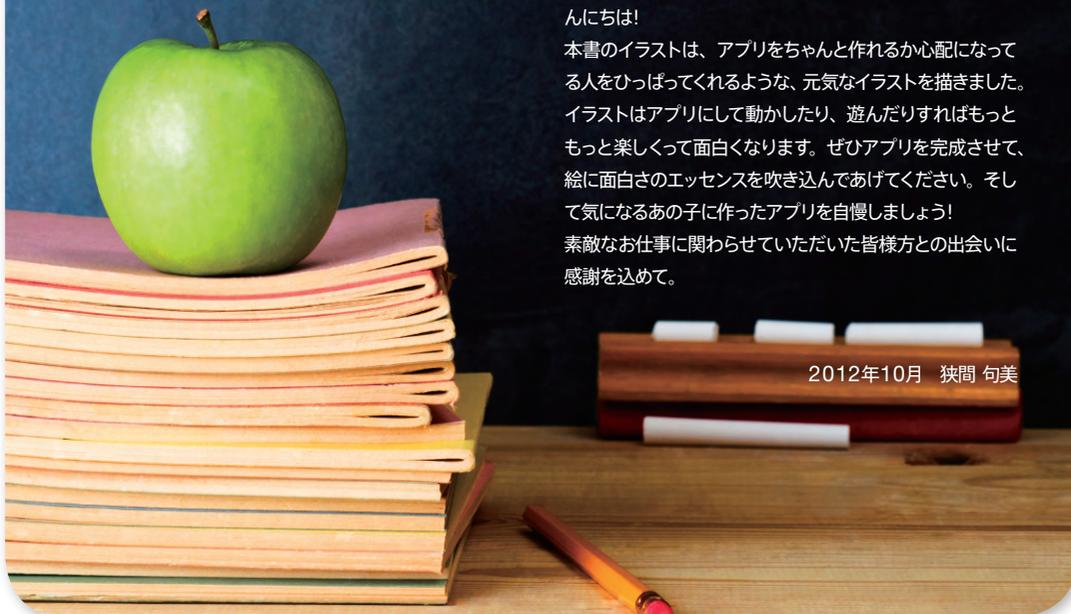
「とはいえ、何から何まで作るの大変だなあ」という人の
ために、サンプルファイルも用意しています。サンプルファ
イルを使えばラクに作れてしまうんですけど、それでも完成
できる喜びはそれなりに味わえます。
ぜひ、「自分の手でアプリを完成させる喜び」を味わって下
さいね。
(サンプルファイルそのままのアプリはAppStoreに公開
したりしないでくださいね。)

最後に、この本を書くにあたって協力して下さった皆さん、
いろいろ無理なことを言っても快く引き受けて書籍に仕上げ
て下さった編集担当の角竹さん、サンプルアプリとしては
もったいないほどかわいいイラストを描いてくれた狭間さん、
ありがとうございました。

2012年10月 森 巧尚

アプリのイラストを描かせていただきました、狭間です。こ
んにちは!
本書のイラストは、アプリをちゃんと作れるか心配になっ
てる人をひっぱってくれるような、元気なイラストを描きました。
イラストはアプリにして動かしたり、遊んだりすればもっと
もっと楽しくって面白くなります。ぜひアプリを完成させて、
絵に面白さのエッセンスを吹き込んであげてください。そし
て気になるあの子に作ったアプリを自慢しましょう!
素敵なお仕事に関わらせていただいた皆様方との出会いに
感謝を込めて。

2012年10月 狭間 句美



本書の読み方 ~カリキュラムについて~

基本

Chapter 1 iPhoneアプリを作るには……

Chapter 1はiPhoneアプリの基本について、一通り説明します。

実習

Chapter 2 はじめてのアプリを作ろう

- Lesson 1 空っぽのアプリを作る
- Lesson 2 文字を表示するアプリを作る
- Lesson 3 簡単なプログラムで動くアプリを作る
- Lesson 4 ボタンを押したら文字が変わるアプリを作る

Chapter 3 ボタンを押して動くアプリを作ろう！

- Lesson 1 おみくじアプリ (その1)
- Lesson 2 おみくじアプリ (その2) ランダムに結果が変わる
- Lesson 3 おみくじアプリ (その3) 結果がじわっと出てくる

Chapter 4 ジェスチャーで動くアプリを作ろう！

- Lesson 1 ねこアプリ (その1)
- Lesson 2 ねこアプリ (その2) いろんなジェスチャーで動くねこ

Chapter 5 時計アプリを作ろう！

- Lesson 1 ずっと動き続ける「デジタル時計」アプリ
- Lesson 2 アナログ「時計」アプリ

Chapter 6 ゲームアプリを作ろう！

- Lesson 1 金魚すくいアプリ (その1) ボイを水につけたり、あげたり
- Lesson 2 金魚すくいアプリ (その2) 金魚を泳かせて、捕まえる
- Lesson 3 金魚すくいアプリ (その3) タイトルとゲームオーバーを作る

Chapter 2からChapter 6までは、「やってみよう」(実習)→「詳しい説明」(解説)の流れで、レッスンが進んでいきます。まず「やってみよう」でサンプルを作成してみましょう。そして、その中で登場したキーワードや、アプリのしくみについて、「詳しい説明」で解説していきますので、理解を深めてください。

Chapterの最後には、アプリの応用例も紹介していますので、挑戦してみてください。

仕上げ

Chapter 7 アプリを仕上げよう

Chapter 7では、アプリの仕上げ方やiPhone 5への対応について解説します。



はじめての アプリを作ろう

この章の目標

- ・ iPhoneアプリを作る流れが分かる
- ・ Xcodeの基本操作が分かる

一番簡単なiPhoneアプリを
作りましょう。



やってみよう

Lesson 1



空っぽのアプリを作る

このLessonの目標

- ・ iPhoneアプリを作る流れがわかる。
- ・ Xcodeの基本操作がわかる。

まず最初に、なんにもしない空っぽのアプリを作ります。

なんにもしないですけど、App Storeからダウンロードするのではなく、自分の手で作るアプリです。自分で作れるんだ、ということを感じてみてください。

それでは、作っていく手順を説明しますね。



真っ白！
だけどこれでも
ちゃんとした
アプリです。



手順

テンプレートから選ぶ

Check!

P.036-040 ではアプリの作り方を学びましょう!

1

Xcodeを起動して、表示される「Welcome to Xcode」のウィンドウから 「Create a new Xcode project」 を選択します。

「Welcome to Xcode」ウィンドウが表示されない場合、Xcodeのメニューから [File > New > new Project] を選択してください。

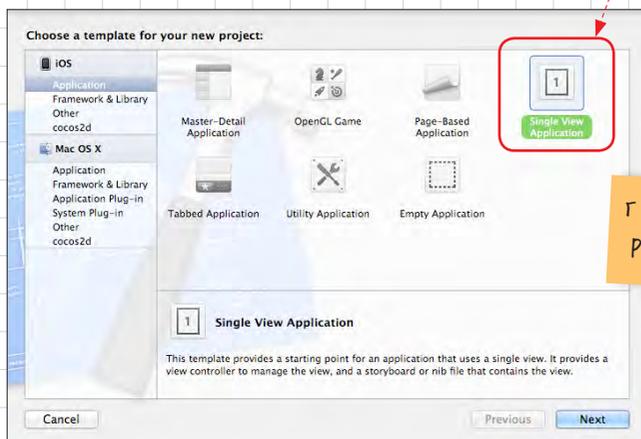


Lesson

1

空っぽのアプリを作る

[Choose a template for your new project:] ウィンドウが表示されます。テンプレート一覧から、「Single View Application」 を選択しましょう。[Next] ボタンを押して次に進みます。



「テンプレート」については P.041へ!

プロジェクトを作る

2

[choose options for your new project:] ウィンドウで、以下のように入力・設定します。

Product Name: hello1

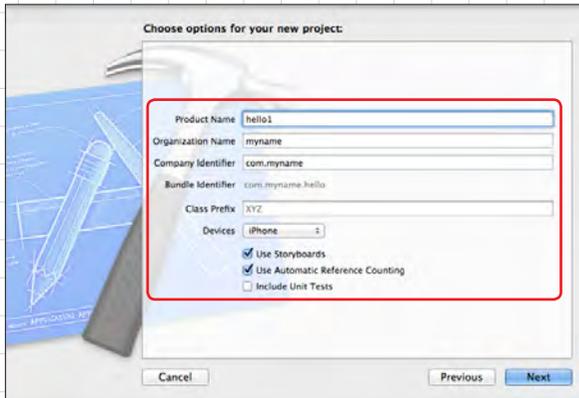
Organization Name: myname

Company Identifier: com.myname

Devices: iPhone

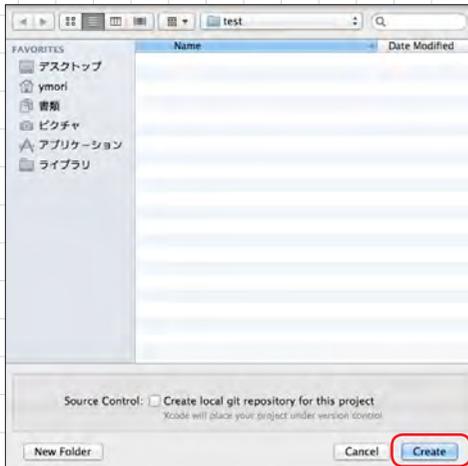
Use Storyboards: チェックを入れる

Use Automatic Reference Counting: チェックを入れる



「プロジェクト」については
P.043 へ!

[Next] ボタンを押すと、保存ダイアログが現れますので、[Create] ボタンを押して保存します。[Product Name] で入力した名前のフォルダが作られます。そして、Xcodeウィンドウが開きます。これが先ほど作った「hello1」プロジェクトの中身で、ここでiPhoneアプリを作っていくことになります。



Createボタン

手順

Run で実行する

3

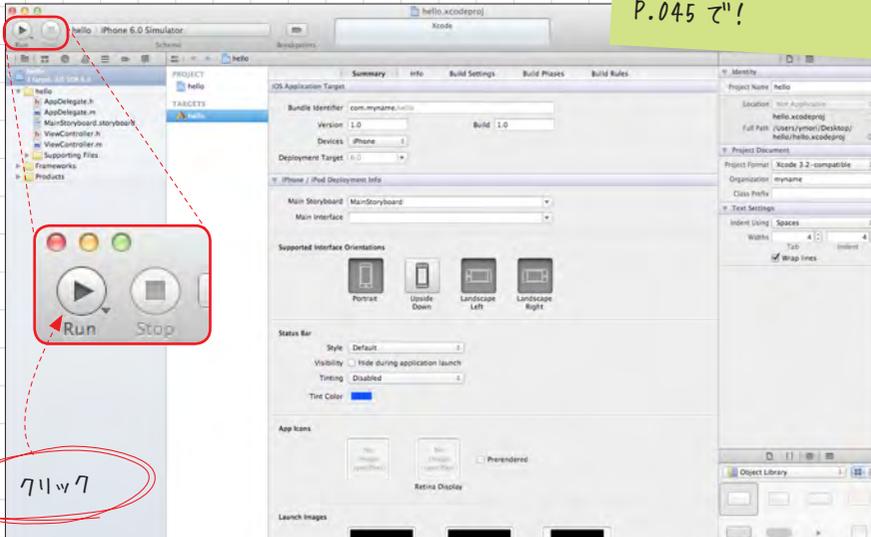
ここからXcodeウィンドウで、画面に部品を配置したり、プログラムを作ったりしていきます。

しかし最初のレッスンでは空っぽのアプリを作りますので、このままウィンドウ左上にある [Run] ボタンを押しましょう!

アプリが作られはじめ、しばらくすると

iOSシミュレータが自動的に起動します。

iOSシミュレータについては
P.045 だ!



iOSシミュレータの画面は……、真っ白です。空っぽのアプリ。

このアプリはまだ空っぽですけど、アプリ開発はここから始まりますよ。

完成!



Lesson

1

空っぽのアプリを作る

アプリを終了するには、[Run] ボタンの横の [Stop] ボタンを押します。



まとめ

意外に簡単だったでしょう？ Chapter1でも簡単に説明しましたが、このように、

1. テンプレートを選ぶ
2. プロジェクトを作る
3. 画面に部品を配置したり、プログラムを作ったりする (ここではやりませんでした)
4. iOSシミュレータでテストする

というのが、iPhoneアプリ作成の基本的な流れとなります。この後も、基本的にはこの手順で作成していきますので、覚えておいてください。

さて、手順の中で、いくつかの耳慣れない言葉が出てきましたね。「テンプレート」「プロジェクト」「iOSシミュレータ」……これらについて、次に解説していきましょう。

詳しい説明

ここから、先ほどまでの手順で出てきた用語について、説明していきます。

Keyword

1

テンプレート

Keyword

3

iOS シミュレータ

Keyword

2

プロジェクト

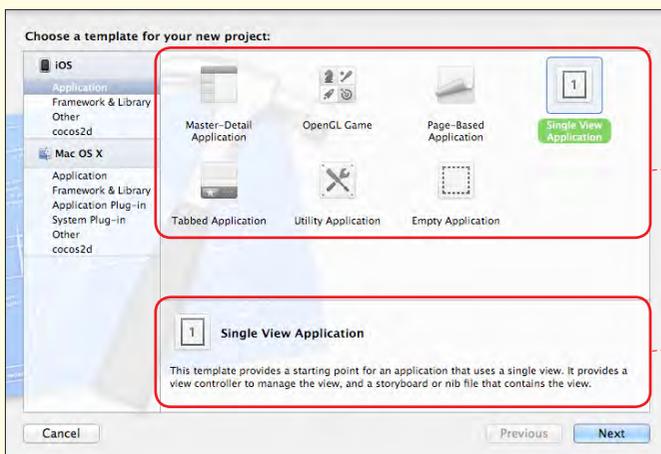
Keyword

1

テンプレートってなに？

Xcodeでは、自分が作りたいアプリに一番近い「テンプレート」を選んで、作っていきます。プロジェクトを作るとき、最初にダイアログが現れます。このとき選ぶのが「アプリのテンプレート」です。

アプリは、まったく何も無いところから作り出すのではなく、あらかじめ用意されたテンプレートの中からどれかひとつ、一番近いものを選んで作っていきます。



テンプレートの一覧が表示されています。

上で選択したテンプレートの説明です。英語ですが...

テンプレート一覧

Single View Application

ビュー(画面)1枚で表示するテンプレートです。
一番シンプルなので、ちょっとしたアプリや、テストをするときなどは、これをよく使います。



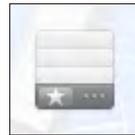
Utility Application

ボタンをタップすると、クルッとひっくり返るアプリ用のテンプレートです。
ビュー(画面)が表裏の2枚あります。



Tabbed Application

画面下のタブをタップすると、画面が切り替わるアプリ用のテンプレートです。
ビュー(画面)が複数枚あります。



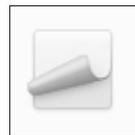
Master-Detail Application

メールアプリのように、並んだリストから一つを選ぶと、画面がスライドして詳細が表示されるアプリ用のテンプレートです。



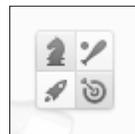
Page-based Application

電子書籍のように、ページをめくって表示するアプリ用のテンプレートです。
Xcode 4.2からできたテンプレートで、実行環境はiOS 5以降です。



OpenGL Game

OpenGL ESという、2Dや3Dのグラフィックスを表示する機能を使ったテンプレートです。凝ったゲームを作るときなどに使います。

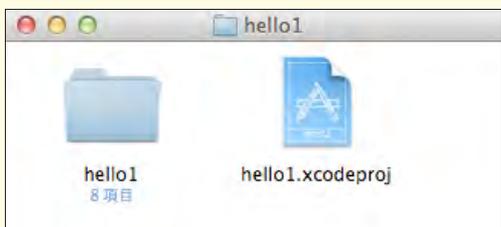


プロジェクトってなに？

プロジェクトとは、アプリを作るのに必要なプログラムや絵などを入れる「フォルダ」のことです。

アプリは、「プログラムファイル」や「設定ファイル」や「画像ファイル」などたくさん
のファイルを使って作ります。

これらを、プロジェクト名のフォルダを作ってそこにまとめて入れて使い、このひと
まとめにしたものを「プロジェクト」と呼びます。



先ほど作った「hello1」
プロジェクトのフォルダ

プロジェクト作成ダイアログ

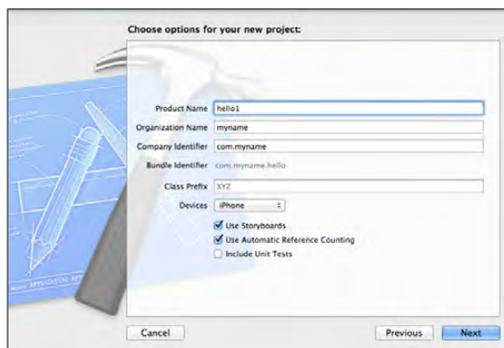
プロジェクトを作るときは、いろいろな設定をします。

Product Nameってなに？

このプロジェクトの名前です。

そのままアプリの名前としても使
われます。

ただし、iPhone上に表示されるア
プリ名はあとから別に指定するこ
ともできます。



Organization Name ってなに?

作った組織の名前です。制作者や制作した会社の名前を入力します。
公開しないでテストするだけだったら「myname」など、適当な名前がかまいません。

Company Identifier ってなに?

公開しないでテストするだけだったら「com.myname」など、適当な名前がかまいません。でも、実際にApp Storeで公開するときは、他のアプリと混乱したりしないように、作った人（会社）の名前などを使って、世界でたった一つの名前で指定します。

ただ、人名や会社名では同姓同名の場合もあるので、持っているURL（これだったら1つしかないですからね）を逆さまにひっくり返して使うことが推奨されています。例えば、URLが「aaaaa.com」だったら「com.aaaaa」と指定します。

Devices ってなに?

作るアプリが、iPhone用アプリなのか、iPad用アプリなのか、Universal (iPhoneともiPad両方用) のアプリなのかを選びます。

それだったら両方用のUniversalがいいように思いますが、そのかわりiPhone用とiPad用の画面を2種類作るのができます。

Use Storyboards ってなに?

画面を作るとき、新しい方式で作るかどうかのチェックです。

古いxib方式（チェックなし）にするか、新しいStoryboards方式（チェックあり）にするかで変わります。

複数画面のアプリをつくる時、xib方式では画面を1つずつ別々に作っていましたが、Storyboards方式だと同時に複数画面を作れてしかも切り換え方まで指定できるようになりました。

本書では複数画面のアプリは作りませんが、チェックあり（Storyboards方式）で行います。

Use Automatic Reference Counting ってなに?

かしこいメモリ管理をするかどうかのチェックです。

チェックしておく、これまでややこしかったメモリ管理をXcodeがかしこく行ってくれる機能がオンになります。

本書ではチェックありで行います。

iOS シミュレータってなに？

iOSシミュレータは、iPhoneそっくりのアプリです。
これでテストをします。

Xcodeでアプリを作って[Run] ボタンを押すと、自動的に起動するiPhoneそっくりのアプリです。この上で作ったアプリが起動してテストすることができます。

メニューの[ハードウェア > デバイス] から、iPhone (Retina 3.5-inch) <iPhone 4S>やiPhone (Retina 4-inch) <iPhone 5>やiPadなどを選んで切り替えることもできます。

本書では、iPhone 4S以前のアプリを制作していきます。
iPhoneやiPhone(Retina 3.5-inch)を使用してください。



iPhone



iPhone Retina 3.5-inch

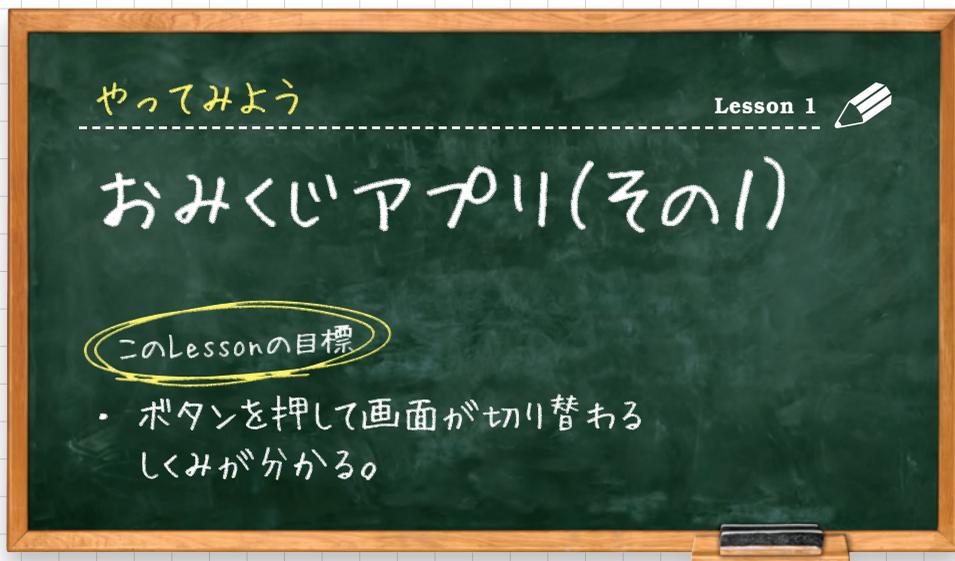


iPhone Retina 4-inch



iPad

また、横に倒したり、ソフトキーボードが出てきたり、ホームボタンをクリックするとアプリが終了したり、と本物のiPhoneそっくりの動きをします。



Chapter2では「ボタンを押したら、文字が変わるアプリ」を作りました。Chapter3では「ボタンを押したら、おみくじをするアプリ」を作りますよ。しかし、いきなり「完成版のアプリ」を作るのはちょっと大変なので、最初は「ボタンを押すと、大吉が表示される」というだけの試作バージョンのアプリを作ります。

「おみくじを引く」ボタンを押したら、「おみくじの結果」を表示するアプリです。ただし、結果は「大吉」の1種類だけ。まずは「画面を切り替えるしくみを、どうやって作るか」に集中してみましょう。

いろいろな方法がありますが、ここでは2枚の絵を使って切り換える方法で作ります。

「タイトル」と「おみくじ結果」の2枚の絵を用意して、最初は「タイトル」だけを表示。「おみくじを引く」ボタンを押したら、上に「おみくじ結果」を表示させます。



完成図




画像ファイル



ViewController

Image View

Image View - Displays a single image, or an animation described by an array of images.



Round Rect Button

Round Rect Button

Round Rect Button - Intercepts touch events and sends an action message to a target object when it's tapped.



Image View

Image View

Image View - Displays a single image, or an animation described by an array of images.



Round Rect Button

Round Rect Button

Round Rect Button - Intercepts touch events and sends an action message to a target object when it's tapped.

アプリ画面に使う部品

使う部品
リスト

ボタンを押すと、画像を表示する。



プログラムのしくみ

作る手順

- 1 プロジェクトを作る
- 2 絵を読み込む
- 3 ステータスバーを隠す
- 4 ライブラリから部品を並べて画面を作る
- 5 部品とプログラムをつなぐ
- 6 プログラムを書く
- 7 Runで実行

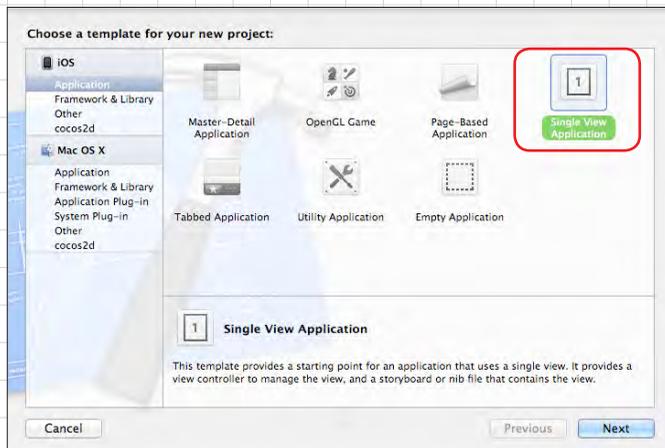
どんなしくみ?(1)についてはP.111で!

手順

プロジェクトを作る

1

Xcodeを起動して、[Create a new Xcode project]を選択し、テンプレートから[Single View Application]を選択しましょう。
[Next] ボタンを押して次に進みます。



以下のように入力・設定します。

Product Name: omikuji

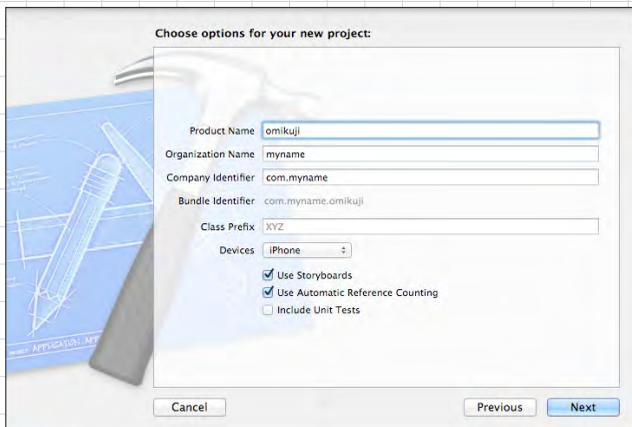
Organization Name: myname

Company Identifier: com.myname

Devices: iPhone

Use Storyboards: チェック

Use Automatic Reference Counting: チェック



[Next] ボタンを押すと、保存ダイアログが現れますので [Create] ボタンを押します。

手順

絵を読み込む

2

絵を用意して、プロジェクトに読み込みます。「タイトルの絵」と「おみくじ結果(大吉)の絵」の2枚です。



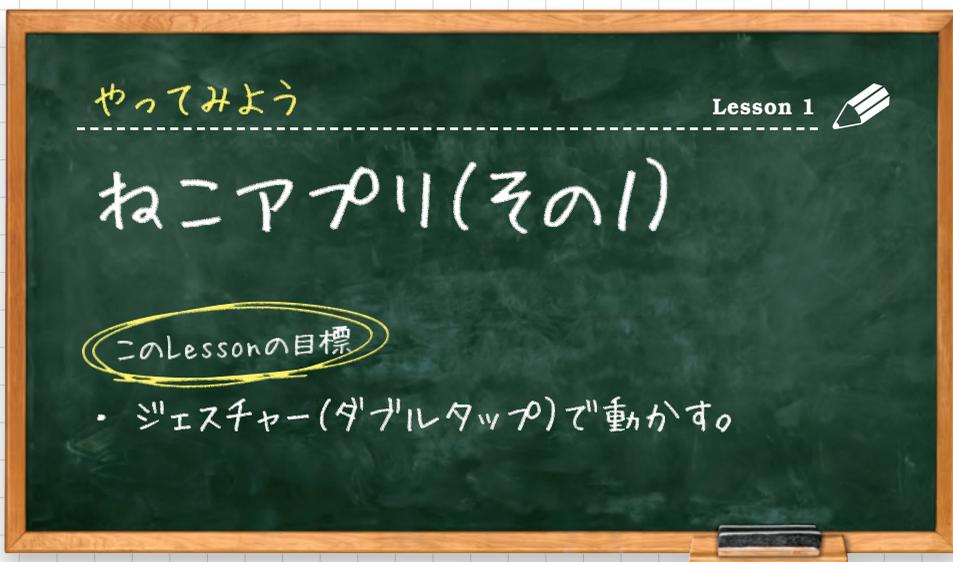
「おみくじタイトルの絵」
(640×960ピクセル) : kuji_top.jpg



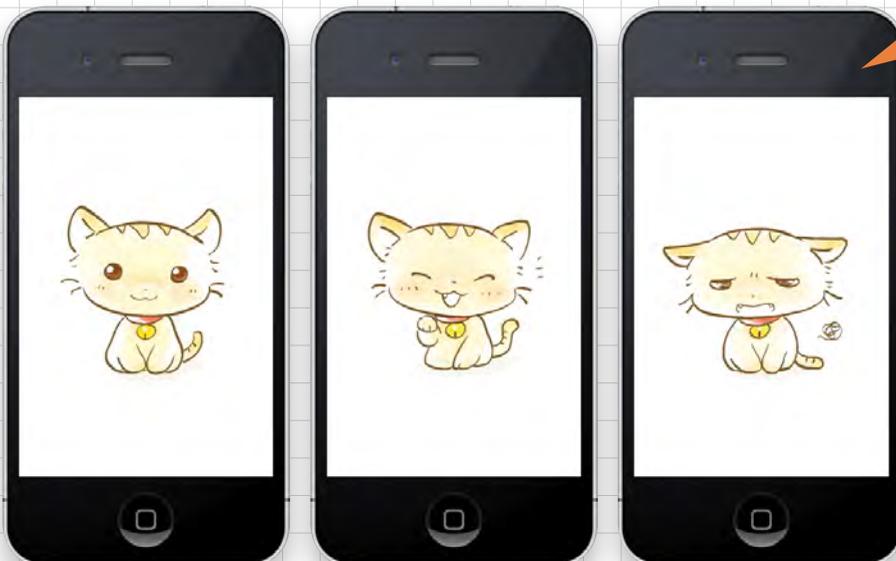
「おみくじ結果(大吉)」
(640×960ピクセル) : kuji_1.jpg

Retina(レティナ)ディスプレイでもきれいに見えるように、高い方の解像度に合わせて用意しておきます。

iPhone 3G/3GSは、320×480ピクセル、iPhone 4/4S(Retina)は、640×960ピクセルです。



ジェスチャーにはいろいろありますが、「ダブルタップ」を使って動くアプリを作ります。画面をちゅんちゅんとタップ(ダブルタップ)すると、ねこがいろいろな表情をしたり、寝たりするんです。



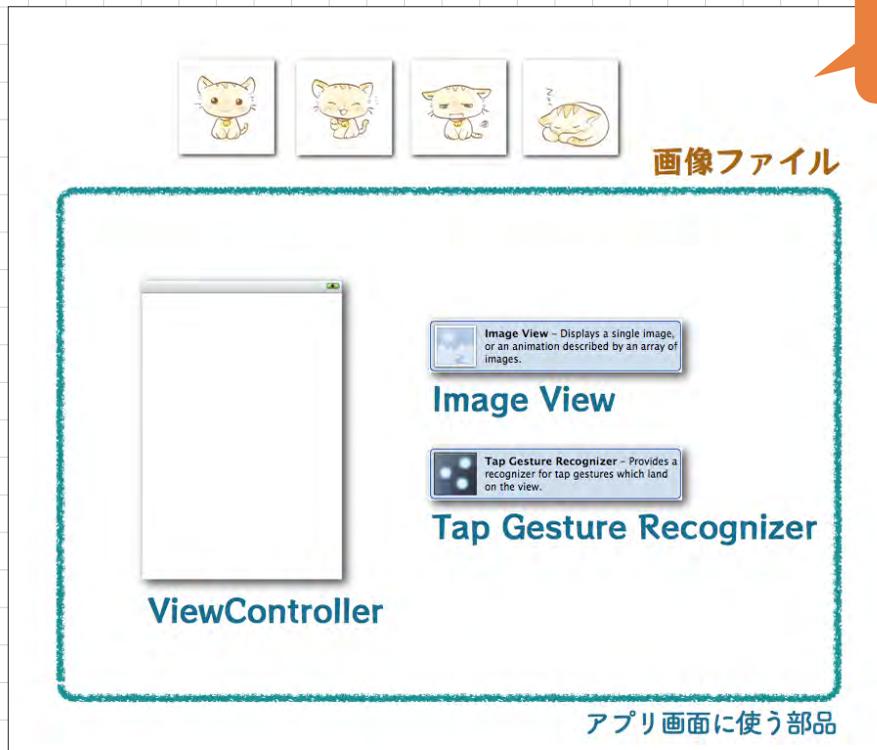
ダブルタップされたかどうかを調べるには、「Tap Gesture Recognizer (タップ・ジェスチャー・レコグナイザー)」という部品を使います。

「レコグナイザー!」なんていう名前なのでどんな難しい部品だろうって思いますが、簡単に言うと「ちょっと変わったボタン」のような部品です。

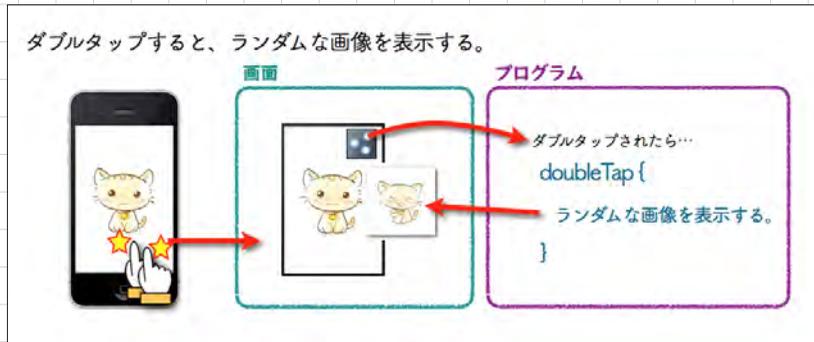
「ボタン」では、ユーザーがボタンを押したとき、仕事を実行できましたよね。

同じように「タップ・ジェスチャー・レコグナイザー」を使うと、ユーザーが画面をタップをしたとき、仕事を実行させることができます。

「ユーザーが何かアクションをしたら、何かを実行する」というしかけなので、「ボタン」も「レコグナイザー」も同じような方法で作ることができます。



「ユーザーがダブルタップをしたら、ランダムな画像を表示する」というしくみで作ります。



プログラムのしくみ

どんなしくみ?(1)についてはP.165で!

作る手順

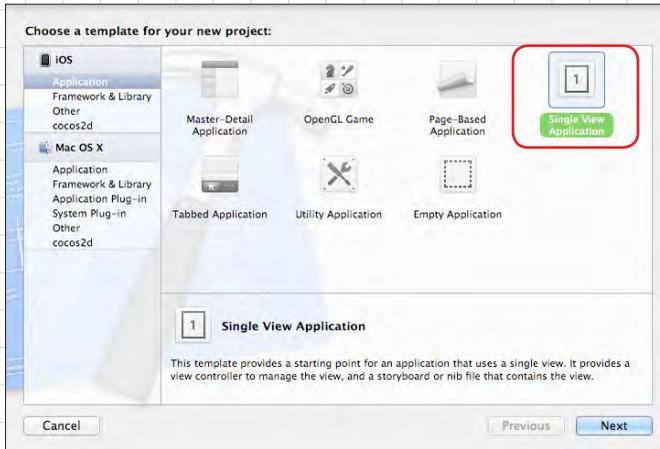
- 1 プロジェクトを作る
- 2 絵を読み込む
- 3 ステータスバーを隠す
- 4 ライブラリから部品を並べて画面を作る
- 5 部品とプログラムをつなぐ
- 6 プログラムを書く
- 7 Runで実行



プロジェクトを作る

1

Xcodeを起動して、[Create a new Xcode project]を選択し、テンプレートから[Single View Application]を選択しましょう。
[Next] ボタンを押して次に進みます。



[Product Name] に「neko」と入力します。

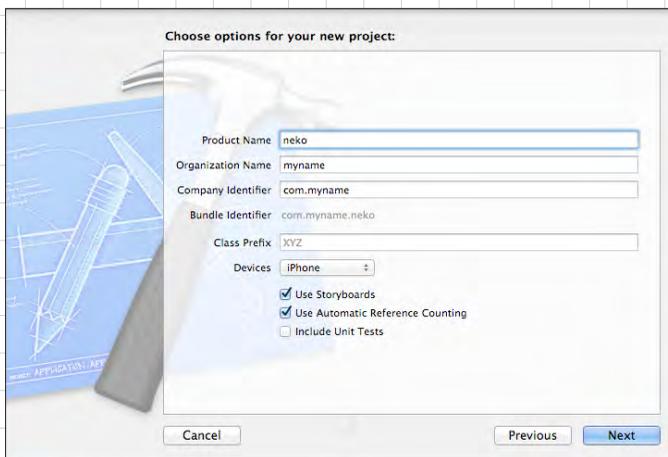
Company Identifier: com.myname

Organization Name: myname

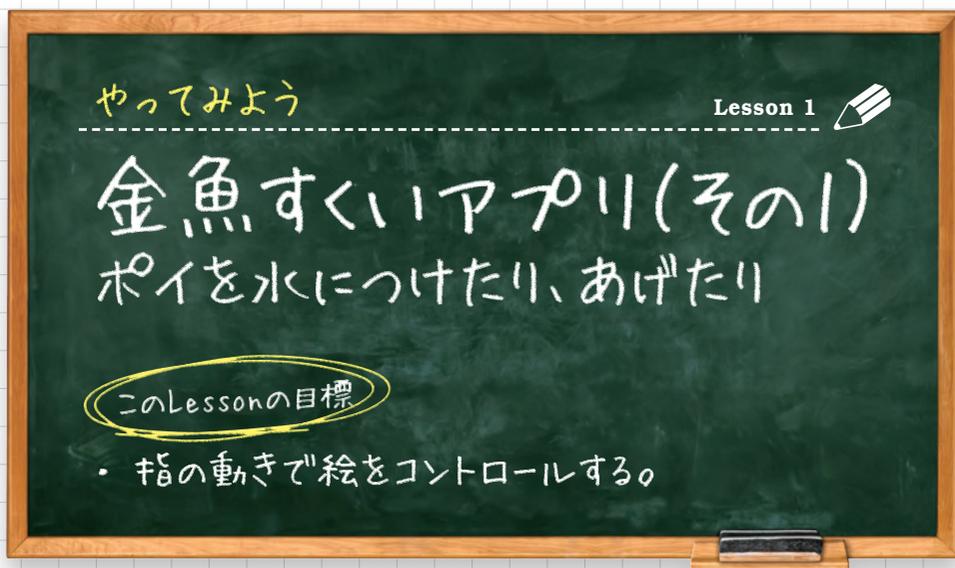
Devices: iPhone

Use Storyboards: チェック

Use Automatic Reference Counting: チェック



[Next] ボタンを押すと、保存ダイアログが現れますので[Create] ボタンを押します。



縁日に行くとよくありますね。

水槽で泳いでいる金魚を、ポイ（紙の膜を丸く貼った道具）ですくうゲームです。

でもそれを、いざアプリで作ろうと思ったら、いったいどこから作ればいいでしょう。

いろいろあって複雑な気がしますが、こういうとき一番大事なのは「観察」です。

「金魚すくい」をぜんぜん知らない人に説明できるぐらいに、ルールやしくみを観察しましょう。アプリを作るとは、何も知らないiPhoneに「金魚すくい」を教えてあげるようなことなのです。

- ・ 金魚がたくさん泳いでいる水槽があります。
- ・ ポイを水槽に入れて、金魚を狙って、下からすくって捕まえます。
- ・ ポイは紙でできているので、そのうち破けてしまいゲーム終了になります。

一度に全部を作ろうとすると大変なので、「少しずつわけて」作ってみましょう。

どんな
アプリ？

「ポイを操作するところだけ」「金魚が泳ぐところだけ」と、部分部分にわけて考えます。

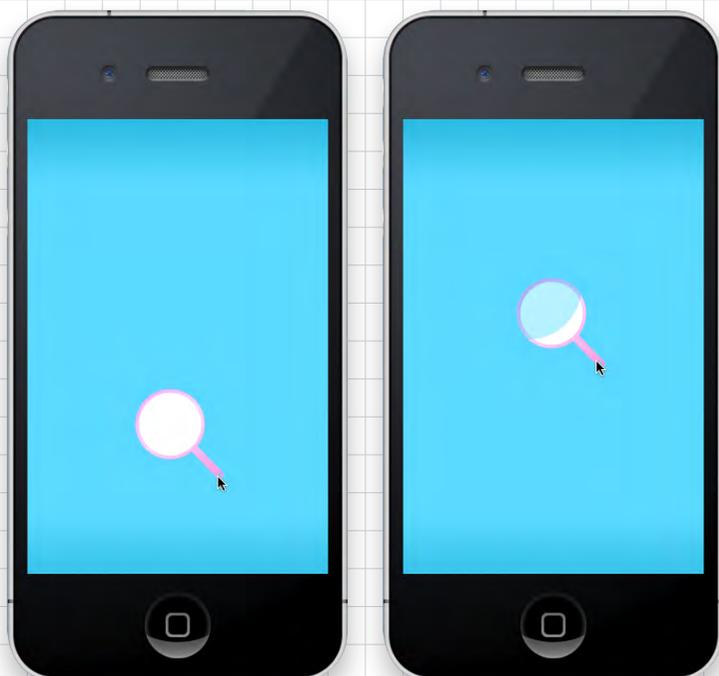
今回のこのアプリでは、以下の3段階に分けて作っていくことにします。

- 1) ポイを水につけたり動かしたりできるようにする(金魚なし)。
- 2) 金魚を泳がせる。その金魚をポイで捕まえる。
- 3) タイトルとゲームオーバーをつけてゲームらしくする。

この分け方や順番に、特に決まりはありません。

「作りやすいところ」から作り始めると、進みやすいので自信がつかますし、「作りたいところ」から作り始めると、面白いのでやる気が出ますよ。

最初は「ポイを水につけたり動かしたりする部分」を作ります。たったこれだけなら、ちょっとできそうな気がするでしょう？



完成図

シミュレータではマウスで操作しますが、iPhoneでは指で操作します。

スマートフォンなので、ポイは指で操作するようにしましょう。

- 画面に指でタッチすると、指先でポイが水につかります。
 - そのまま指を動かすと、ポイが指にくっついて移動します。
 - 指を上げるとポイも水から上がって、すくい上げることができます。
- と、このような動きです。

この「指で操作する3つの動き」は、iPhoneアプリでは基本的な動きです。それぞれ、

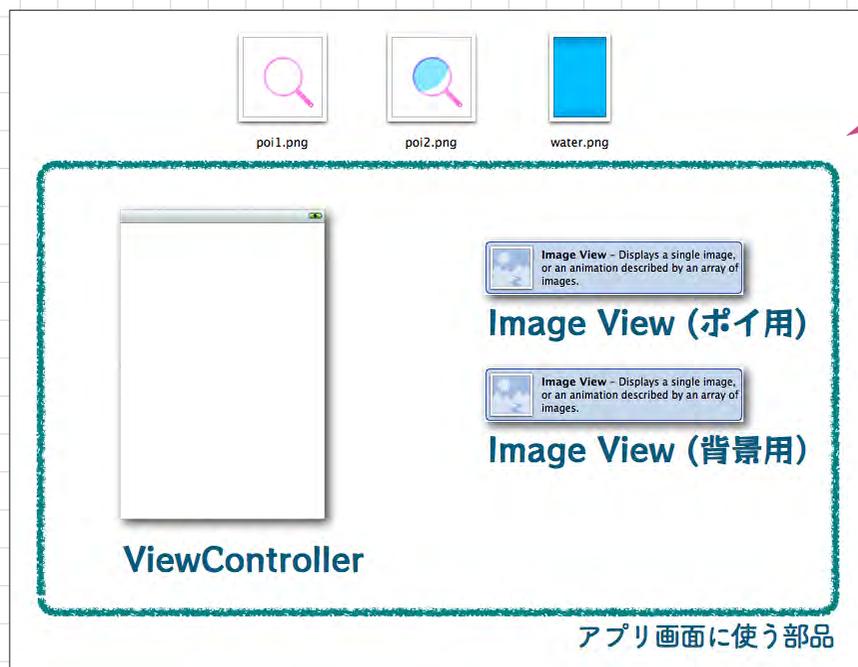
「touchesBegan (タッチしはじめたとき)」

「touchesMoved (タッチして動かしたとき)」

「touchesEnded (タッチしていた指を放したとき)」

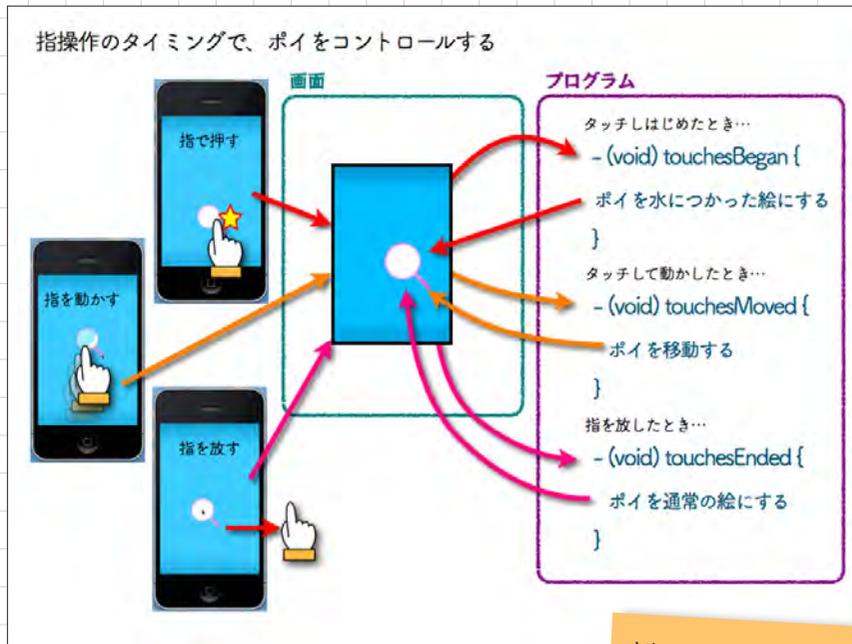
という基本的な機能が用意されています。

このそれぞれの指操作のタイミングで、ポイをコントロールします。



使う部品
リスト

「ユーザーがタッチしたら、ポイを水につける。指を動かしたら、ポイを動かす。指を上げたら、ポイを水から上げる。」を作ります。



プログラムの
しくみ

どんなしくみ?(1)
については P.236 へ!

作る手順

- 1 プロジェクトを作る
- 2 絵を追加する
- 3 ステータスバーを隠す
- 4 ライブラリから部品を並べて画面を作る
- 5 部品とプログラムをつなぐ
- 6 プログラムを書く
- 7 Runで実行