cocos2d-x ゲーム開発 実践セミナーブック

八反田 智和（HatchUp）/編著

現場直伝!

©2013-2014 Alim Co., Ltd.
All Rights Reserved.

あのゲームの開発者が講師!

他では聞けない現場直伝の
cocos2d-xのテクニックやTipsが学べる

誌上セミナー、開講!

講師 清水 友晶（TKS2）/ 有賀 之和・桜庭 洋之・堅田 瞬（フルセイル）/ 尾崎慎一・深野 真人（Goodia）/ 杉山 浩・小林 伸隆（エイリム）/ 木内 智史（BEFOOL）

マイナビ

現場直伝!

cocos2d-x ゲーム開発 実践セミナーブック

八反田 智和(HatchUp)/編著

マイナビ

◉ 本書のサポートサイト

本書のサンプルファイルや補足情報、訂正情報などを掲載してあります。適宜ご参照ください。

http://book.mynavi.jp/support/pc/4888/

◎ **本書は、cocos2d-xのTips、テクニック書です。**
基本的な導入方法や環境構築、制作の基本の流れなどは解説しておりませんので、
別途他の解説書をご参照ください。

◎ **本書は、cocos2d-xで実際に開発されたゲーム事例を解説しているため、開発当時に使用したcocos2d-xのバージョンに基づいた解説内容となっています。そのため、項目によって、使用しているcocos2d-xのバージョンが異なっております。あらかじめご了承ください。それぞれの章でのcocos2d-xのバージョンは、章の冒頭に記載してあります。**

◎ **本書は2014年6月段階での情報に基づいて執筆されています。**
本書に登場するソフトウェアのバージョン、画面、機能、URL、製品のスペックなどの情報は、すべてその原稿執筆時点でのものです。
執筆以降に変更されている可能性がありますので、ご了承ください。

◎ 本書に記載された内容は、情報の提供のみを目的としております。
したがって、本書を用いての運用はすべてお客様自身の責任と判断において行ってください。

◎ 本書の制作にあたっては正確な記述につとめましたが、
著者や出版社のいずれも、本書の内容に関してなんらかの保証をするものではなく、内容に関するいかなる運用結果についてもいっさいの責任を負いません。あらかじめご了承ください。

◎ 本書中の会社名や商品名は、該当する各社の商標または登録商標です。
本書中では™および®マークは省略させていただいております。

はじめに

本書は、cocos2d-xによって開発されたスマートフォンゲームアプリの開発実例や、開発現場で重宝するノウハウを扱う日本国内で最初の書籍です。

本書では、現在、日本国内外で活躍していらっしゃるゲーム会社各社の代表ゲーム事例を複数集め、ていねいに解説しています。ゲームジャンルも、カジュアルゲームから本格的なソーシャルゲームまで、幅広くカバーしています。また、内容的にも、cocos2d-xの開発ノウハウやTipsにとどまらず、ゲームコンセプトや企画、UIデザインなど幅広く実践的な情報を盛り込んでいます。
中には、業界内で有名なヒットシリーズや売上ランキングで国内トップのタイトルも盛り込まれており、開発をした会社の体制や内情なども垣間見える、生々しい内容となっていますので、業界人必見です。
スマートフォンゲームアプリ開発現場の企画・開発職の方をはじめ、これからゲームアプリ開発を始めようと考えていらっしゃる学生さんや、一般のアプリ開発者の方々にもわかりやすいよう、具体例を交えて解説していますので、ぜひお読みいただければと思います。

なお、cocos2d-xのテクニックやノウハウに特化したため、本書ではcocos2d-xの開発環境構築や、基本的なプログラミングの流れについては解説しておりません。
基礎から本格的にcocos2d-xを学習したい方は、本書Session1執筆担当である清水 友晶さんが執筆された『cocos2d-xによるiPhone/Androidアプリ プログラミングガイド』（マイナビ刊）などの解説書を手に取るとよいでしょう。

cocos2d-xは、日本を含む東アジア4カ国（日本、中国、韓国、台湾）や、欧米など、世界中で急速に普及し、大手ゲーム企業から個人ディベロッパーまで、導入事例が加速度的に増えています。いまや、すべてのスマートフォンゲームクリエイターにとって外すことのできない、習得必須の開発環境といって過言ではないでしょう。
本書を手に取り、イメージを膨らませ、実際のあなた自身のゲーム開発に存分に活かしていただければ幸いです。

2014年6月
八反田 智和

Contents

Session 1

【基本編】
ゲーム開発と
cocos2d-xの概要

010

講師：
清水 友晶（TKS2）

1-1 ゲーム開発のターゲットOSと開発環境 012
日本におけるスマートフォンのOS別販売シェア 012
ゲーム開発の問題 013
クロスプラットフォーム開発 015
1-2 cocos2d-xと他のゲーム向けミドルウェアの比較 018
cocos2d-x、Corona SDK、Unityの比較概要 018
価格 019
開発言語 020
開発環境 020
UIエディタ 022
公式サポート・日本語コミュニティ 025
対応プラットフォーム 026
1-3 cocos2d-xのメリットとデメリット 028
cocos2d-xを利用した時のメリット 028
cocos2d-xを利用した時のデメリット 029
1-4 cocos2d-xのトレンド 030
cocos2d-xの国内トレンド 030
cocos2d-xの海外トレンド 032

Session 2

【事例編】
カジュアルゲーム
開発に見る
cocos2d-x
活用テクニック

036

講師：
有賀 之和・桜庭 洋之・堅田 瞬
（フルセイル）

2-0 cocos2d-xの魅力 040
cocos2d-xを導入してみて 040
開発者の視点 041
今後の戦略 041
2-1 事例1：アニメーションによる演出方法 042
フレームアニメーションにおけるテクスチャアトラスの利用 043
テクスチャアトラスおよびデータファイルの作成方法 044
フレームアニメーションの実行方法 047
オープニングのアニメーション 048
テキストの表示 048
キャラクターの登場 050
干支の玉が飛んでいくアニメーション 051
2-2 事例2：背景絵のスクロールアニメーション 053
背景のスクロールアニメーション 053
背景スプライトの作成 054
背景スプライトのアニメーション 056
移動にもっと臨場感をもたせる 057
キャラクターを走らせる 060

無限にループさせる 062
2-3 事例3：アプリ内データの使用方法 063
オブジェクト同士のデータ管理 064
広告の配信方法 066
アプリ内のデータ保存 069
2-4 終わりに：cocos2d-xに期待すること 076

Session 3

【事例編】カジュアルゲームで学ぶcocos2d-xゲームプログラミングテクニック

078

講師：
尾崎 慎一・深野 真人
(Goodia)

3-0 cocos2d-xの魅力とアプリ制作ワークフロー 082
cocos2d-xを選んだわけ 082
cocos2d-xを導入してみて 083
今後の戦略 085
Goodiaでのアプリ実装の流れ 086
3-1 プレイヤーの動き（操作） 087
タップ 088
ボタン 090
スワイプ 092
フリック 095
センサー 098
ジョイスティック 100
3-2 自キャラクターの動き（移動） 103
cocos2d-xの関数を使った処理 103
スケジュールを使った処理 104
3-3 障害物と敵キャラクターの動きについて 109
敵キャラクターの動きについて（近づく） 110
敵キャラクターの動きについて（遠ざかる） 112
敵キャラクターの動きについて（上下運動、左右運動） 112
3-4 プレイヤーの動き（ジャンプ） 116
cocos2d-xの関数を使った処理 116
スケジュールを使った処理 117
3-5 オブジェクト同士の当たり判定 122
自キャラクターと敵キャラクターの幅、高さを使った当たり判定 122
自キャラクターと敵キャラクターの距離を使った当たり判定 123
特殊な当たり判定について（複数の当たり判定） 124
3-6 サウンドエフェクト、アニメーション、背景スクロール 127
サウンドエフェクト 127
アニメーション（基礎） 128
アクション速度を変える 130

メニューにアニメーションを設定する 131
キャラクターに連動しないスクロール（流れる雲背景） 133
キャラクターに連動するスクロール（ビルの背景） 137
3-7 レベルデザイン 139
3-8 その他 142
スプライトを回転させた場合の当たり判定 142
ベジェを利用した曲線運動 145
ランダムを応用した組み合わせ 147

Session 4

【事例編】
ブレイブ フロンティアに学ぶcocos2d-x「演出」テクニック
152

講師：
杉山 浩・小林 伸隆
（エイリム）

4-0 ブレイブ フロンティアのリリースと反響 154
日本での状況 154
海外での状況 156
4-1 ブレイブ フロンティアの開発体制について 158
開発期間について 158
開発メンバーについて 158
4-2 cocos2d-xの魅力 159
cocos2d-xを選んだわけ 159
cocos2d-xを導入してみて 160
開発中に苦労した点 162
使用ツール 163
4-3 ブレイブ フロンティアのこだわり 166
バトルはとにかく派手に！ 166
通信を少なく！ 167
今後の戦略 168
4-4 TIPS1：戦闘時のタッチ、長押し、スライド処理 169
4-5 TIPS2：画面演出 173
4-6 TIPS3：描画の高速化 176
4-7 TIPS4：通信 178

Session 5

5-1 アニメーション エトセトラ 183
アニメーションの種類 183
Sprite操作系のAction 183
変化を操作するAction 184
アニメーションの再生を制御するAction 185
5-2 アニメーションGIFに関して 187
アニメーションGIFとは？ 187

Session 5

【Tips編】
制作を効率化!
cocos2d-xの
基本&実践Tips

182

講師:
木内 智史(BEFOOL)

ImageMagickのインストール 187
パターン① Animation::addSpriteFrameWithFileを活用する 188
パターン② Animation::addSpriteFrameを活用する 189
5-3 タップ処理に関して 191
タップ処理事始め 191
単純なSpriteタップ判定 192
ふたつ以上重なる場合のSpriteタップ判定 193
ロングタップの判定 195
ダブルタップの判定 199
5-4 ドラッグ アンド ドロップを実装してみる 201
つかむ 201
落とす 202
5-5 HTTP通信に関して 205
HttpRequest 205
画像ファイルの保存 206
JSONを扱ってみる 207
XMLを扱ってみる 210
5-6 Loading Barを実装してみる 212
HttpRequest編の応用 212
ローディングバー部分の実装 212
実装したローディングバーの活用 214
5-7 変数の永続化に関して 217
変数の永続化とは? 217
プリファレンスによる保存 217
ValueMap(旧Dictionary)による保存 218
SQLiteによる保存 220
5-8 物理エンジン 223
物理エンジンとは? 223
重力場の設定 223
物体の設定 224
ジャンプ 226
衝突 226
PhysicsBodyに設定できる項目 228
5-9 Spine(ボーンアニメーション) 229
Spineとは? 229
ボーンセットアップ 229
ボーンアニメーション 230
cocos2d-xからアニメーションの初期化 231

アニメーションの呼び出し 232
ボーンの操作 233
アニメーションからのイベント通知を受け取る 234
5-10 動画の再生 237
cocos2d-xでの動画再生手段 237
iOS向けの自前実装 237
Android向けの自前実装 242
ui::VideoPlayerを先取りする 248
5-11 WebViewの導入 250
WebViewとは？ 250
GREE Platform SDK for Cocos2d-x 250
Androidへの対応 253
注意点 255
5-12 Luaの利用 256
Luaとは？ 256
Luaで開発するメリット 256
Luaで開発するデメリット 257
サンプルから学ぶ、cocos2d-xとLuaの連携 258
C++からLuaへのアクセス 260
Luaスクリプトの難読化 262
5-13 JS Binding 264
JS（JavaScript）とは？ 264
JavaScript Bindingで開発するメリット 265
JavaScript Bindingで開発するデメリット 265
サンプルから学ぶ、cocos2d-xとJavaScriptの連携 265
5-14 CI（Continuous Integration） 268
CI（Continuous Integration）とは？ 268
TDD（Test Driven Development）という手法 268
テストを書く準備 269
テストを書く 271
Travis CIを利用してCIを実践する 272

TechBuzzによるcocos2d-x勉強会・ハンズオンと、cocos2d-xユーザー会 275
INDEX 277

Session 1

【基本編】ゲーム開発とcocos2d-xの概要

Agenda　このセッションの内容

- 1-1　ゲーム開発のターゲットOSと開発環境
- 1-2　cocos2d-xと他のゲーム向けミドルウェアの比較
- 1-3　cocos2d-xのメリットとデメリット
- 1-4　cocos2d-xのトレンド

Session **1**

Title
ゲーム開発とcocos2d-xの基本

Summary

このセッションでは、ゲーム開発用環境のcocos2d-xの実務での現状、業界動向を現場の目で紹介します。多プラットフォーム対応のスマートフォン向けゲーム開発環境としてのcocos2d-xの位置付けを、他のミドルウェアと比較しながらメリットデメリットを考えます。また国内外の状況も確認します。

Level
初級

Genre
概要編

Skill
- スマホゲームの開発状況
- ミドルウェアの比較
- cocos2d-xの開発環境

cocos2d-x Version
-

Profile プロフィール

清水友晶

cocos2d-x.jp代表、株式会社TKS2代表取締役。
主にiOS・Android・Windows Phone 8アプリを開発。日本cocos2d-xユーザー会「cocos2d-x.jp」代表。東京を中心に、名古屋・福岡・鹿児島など各地でcocos2d-x勉強会やハンズオンセミナーの講演を開催している。

ブログ　http://tks2.net/memo/
過去の講演資料　http://www.slideshare.net/doraemonsss/
Facebookグループ　https://www.facebook.com/groups/cocos2dxjp

Column

筆者から見た株式会社 HatchUp について

筆者は、原稿執筆時（2014年5月）までにcocos2d-xの勉強会やハンズオンで50回以上講演しましたが、そのほとんどが株式会社HatchUpが運営するTechBuzzというイベントの元で開催されたもので、全面的にサポートしていただきました。
またこれがきっかけで日本cocos2d-xユーザ会を立ち上げることになり、cocos2d-xの開発元であるChukong Technologies社とも交流を持たせていただいております。
その日本cocos2d-xユーザ会のコミュニティであるcocos2d-x.jp communityは、公式フォーラムを除けば世界で一番大きなコミュニティに成長しており、Chukong Technologies社の開発者も参加しています。実は、この日本cocos2d-xユーザ会を運営しているのも株式会社HatchUpです。ある意味、日本におけるcocos2d-xはHatchUpにより支えられていると言ってもいいでしょう。

http://www.socialtoprunners.jp

本格的パズルゲームでありながら、英単語（スペル）を学べる。

キャラクターは、「ビックリマン」で知られるグリーンハウスが描き下ろしている。

英単語の学習画面。英語を学ぶとプレイヤーも成長していく仕組みになっている。

株式会社ロノトによりリリースされた「タワーオブスペルズ」は、本格的パズルゲームでありながら、英単語（スペル）を学べる画期的なゲームです。100体を超える全キャラクターは「ビックリマン」を世に送り出したグリーンハウスに描き下ろしていただきました。パズルバトルでストーリーを進めながら、魅力的なキャラクターたちを育成し、さらにスペルリストを増やして英語を学んでいくことにより、プレーヤー自身も成長していくゲームです。

「タワーオブスペルズ」は、cocos2d-x v3.0により開発が進められました。cocos2d-xを利用していたため、パフォーマンス面にそこまで力を入れずとも軽快な操作が実現されています。グリーンハウスが描き下ろしたキャラクターは重要なものですので、サードパーティ製のツールにより画像の暗号化を行いましたが、これもcocos2d-xがサポートしているので導入は非常に容易でした。

またcocos2d-xがオープンソースであったことから、利用しやすいようにクラスの改良や機能の追加を行いました。どれだけ売り上げがあっても全機能を無償で利用できるので、企業にとってはノーリスクです。

http://towerofspells.com

タワーオブスペルズ

Ronoto, Inc.

対応機種：iOS

初回リリース日：2014/05/08

ジャンル：ゲーム

iOS

1-1 ゲーム開発のターゲットOSと開発環境

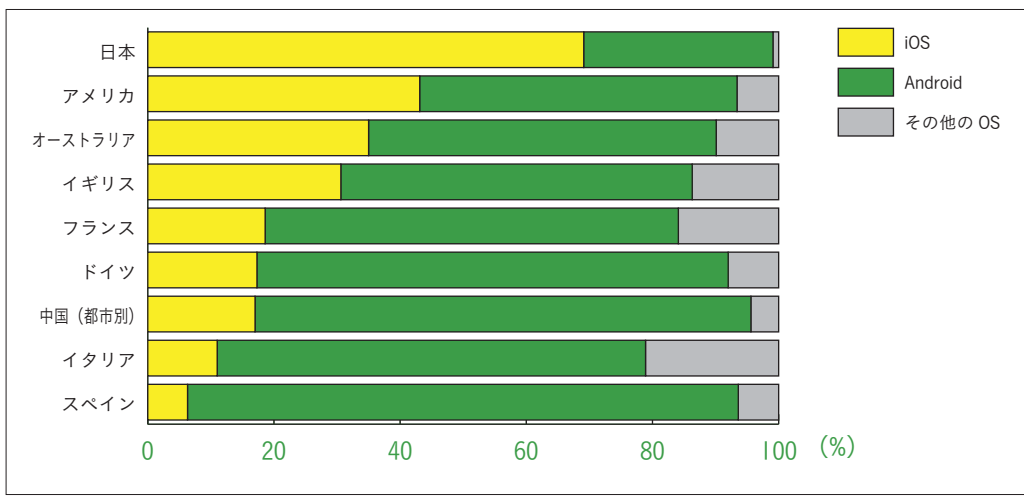
今、あなたがこれまでのスマートフォン市場にないゲームをひらめいたとしましょう。このゲームは、非常に面白いシステムを持っており、市場に出すと100％人気が出ること間違いありません。あなたならどのようにしてこのゲームをリリースしますか？　一言にスマートフォンと言っても、世の中には多くのOSが用意されています。どのOSをターゲットにし、どのように開発を進めればいいか考えてみましょう。

日本におけるスマートフォンのOS別販売シェア

まずはターゲットとなるOSについて考えてみましょう。

表1.1にスマートフォンのOS別販売シェアを示します。日本におけるスマートフォンのOS別販売シェアでは、iOSが69.1％、Androidが30.0％と、この2つのOSが主流となっています。すなわち日本でスマートフォン向けアプリをリリースする場合、iOSに対応したアプリだと全ユーザーの2/3以上の方に利用してもらえるチャンスがあることになります。

表1.1 スマートフォンOS別販売シェア



「カンター・ワールドパネル・コムテック調査」（調査時期：2013年9月から11月）／新規契約又は機種変更したスマートフォン機種のOS別の割合

出典 http://kantar.jp/whatsnew/2014/01/14/NewsRelease_140115_ComTech.pdf

あなたは一所懸命作ったアプリを、日本だけでくすぶらせてもいいのですか？　日本語を英語に翻訳するだけで、世界で利用してもらうことができるでしょう。

世界のスマートフォンのOS別販売シェアを見てみましょう。アメリカでは、iOSが43.1%、Androidが50.3%であり、AndroidがiOSを逆転しています。さらにヨーロッパでは、iOSのシェアは約20%と低く、ほとんどのユーザーがAndroidを利用しています。海外に目を向けるのであれば、Androidを無視することはできないですね。Androidに対応したアプリも作るのであれば、日本のスマートフォンシェアの1/3にあたるAndroidユーザーもカバーすることができます。

これでターゲットとなるOSは、iOSとAndroidに決まりました。これらのOS向けのアプリを作ることを考えましょう。しかし欲を言うと、海外にいるBlackBerryやWindows Phoneのユーザーのために、それぞれのOSでもアプリを作りたいところです。またTizenのような開発中のOSもあり、今後普及する可能性もゼロではないでしょう。スマートフォンのシェアは1年で大きく変わることもあるので、いざという時のことも考えておかなければいけません。これらのような、iOS・Android以外のOSでの開発についても忘れることなく、頭の片隅に置いておきましょう。

ゲーム開発の問題

次は開発の方法について考えてみましょう。

iOSとAndroid向けのアプリを作ることになりましたが、どのように開発を進めればいいでしょうか？　まずは日本に目を向けて、iOS向けのアプリを先に開発するのがいいでしょうか？　それとも海外でシェアが大きいAndroid向けのアプリを先に開発した方がいいでしょうか？　しかしどちらを先に開発したとしても、そのアプリをリリースし一度市場に出してしまうと、あなたのゲームは非常に人気があるので、すぐにマネされてしまいます。ちなみに、ここで言うリリースとは、iOSであればAppleが用意しているApp Storeへ、AndroidであればGoogleが用意しているGoogle Playストアへアプリを公開することを指しています。

図1.1　iOSのApp Store

図1.2　AndroidのGoogle Playストア

例えば、あなたがiOS向けアプリの開発終了と同時にリリースしたとしましょう。そして次にAndroid向けアプリの開発に取りかかります。しかしiOS向けアプリは公開されているので、それを元にゲームシステムをマネすることは容易です。すぐに開発力の高いゲーム会社によってAndroidで似たようなゲームをリリースされてしまうでしょう。馬鹿げたことではありますが、現に多くのゲーム会社がこの問題で悩んでいます。日本国内だけであればオリジナルを重視する傾向が市場にもあるためそれほど大きな問題にはなりませんが、海外ではオリジナルよりもコピーものが売れるということも日常茶飯事です。日本で売れたゲームを海外でもリリースする時、すでにリリースされているコピーゲームの会社を買収し自分たちのゲームが売れるように布石を敷くことも現に行われています。

では他社にマネされないために、開発が終わった一方のアプリをリリースせず、もう一方の開発が終わってから同時にリリースしますか？　確かにマネをされる心配はないですが、ひとつのアプリを作るために倍の時間が必要になります。例えば、iOS向けのアプリ作成に6ヶ月かかったのであれば、Android向けのアプリ作成にも6ヶ月を要するかもしれません。つまり開発開始から1年経たないとアプリをリリースできません。ゲームにも旬がありますので、それを過ぎてしまうとユーザーに見向きされない可能性もあります。ゲームシステムを考えた時は素晴らしいものだったかもしれないですが、1年も経ってしまうと一般常識になっていることもあり得ます。そのため、開発期間はできる限り短くしなければいけません。

クロスプラットフォーム開発

クロスプラットフォーム開発の手法は、今に始まったものではありません。例えば、Windows・Mac・Linux等のクロスプラットフォーム開発は古くから行われています。スマートフォンの普及にともなって、iOSやAndroidにおけるクロスプラットフォーム開発の必要性が高まってきました。と言っても、現在利用されているクロスプラットフォーム手法が突然生まれたわけではありません。

初期の頃のクロスプラットフォーム開発と言えば、ブラウザを利用したものでした。ただし、ブラウザと言ってもiOSのSafariやAndroidのChromeのことではありません。アプリとしてリリースするためには、アプリ内ブラウザと呼ばれるiOSのUIWebViewやAndroidのWebViewを利用する必要がありました。これらのブラウザは、JavaScript言語を読み込むことができるので、これを利用し1ソースでiOSとAndroidアプリの開発が行われていました。日本においては、携帯電話で人気のあったカードゲーム等が多く移植され、今でも遊ばれています。また、現在でも携帯電話とスマートフォンのゲームを同時に開発する時は利用されている手法です。しかし、アプリ内ブラウザを利用しているため、ユーザーの操作に対するレスポンスが悪かったり、画面の構成を含めた通信を行う場合もあるため通信データ量が増えたり、アニメーション処理の遅れによりコンシューマーゲームのように快適なゲームを行うことができなかったり等、多くの問題点も持っています。また別の問題として、アップルによるiOSアプリの審査も年々厳しくなり、アプリ内ブラウザを利用した時の制約も多くなっています。それにより、単純にアプリ内ブラウザを表示しただけのアプリは、審査に通らずリジェクトされてしまうようになりました。

そこで、ゲーム画面のような高い操作性が要求される箇所はOS毎に実装を行い、ヘルプ画面のような早いレスポンスが必要な箇所はアプリ内ブラウザを利用する、複合的な方法が用いられるようになりました。また近年主流となっているフリーミアムモデル（無料アプリとしてリリースし、アプリ内課金にて収益を得る）の場合、アプリ内課金はそれぞれのOSのフレームワーク（iOSの場合はIn-app Purchase、Androidの場合はIn-app Billing）を利用する必要があるため、自ずとこの方式を選ぶことになります。しかしゲームによっては、ゲーム画面が他の画面と密に連携していたり、ゲーム画面がアプリ全体の半分以上の工数を必要としていたり、クロスプラットフォーム開発は簡単ではありませんでした。

現在主流となっているのが、これらの問題を全て解消したミドルウェアを利用したクロスプラットフォーム開発です。これにより、OSが異なってもほぼ全てのソースコードを1ソースで用意できるようになりました。一般アプリを作る場合は、

Titanium Mobile（図1.3）と呼ばれる開発環境がよく利用されます。これを使うとJavaScriptで、iOS、Android、BlackBerry等のアプリを作ることができます。例えば、端末に備わっているカメラを利用する時は、一般的にそれぞれのOSで異なった処理を行う必要があるため、対応OS分のソースコードが必要になりますが、Titanium Mobileが提供しているカメラの機能を利用することにより、OSを気にせず1ソースでカメラの処理を行うことができます。

もちろんゲームアプリを作るためのクロスプラットフォーム開発に対応したミドルウェアもあります。現在一般的に利用されているのはcocos2d-x（図1.4）、Corona SDK（図1.5）、Unity（図1.6）等であり、これ以外にも多くのミドルウェアが存在します。先に紹介したように、現在の主流となっているのはiOS、Androidですので、少なくともこれらのOSをサポートするものがほとんどです。

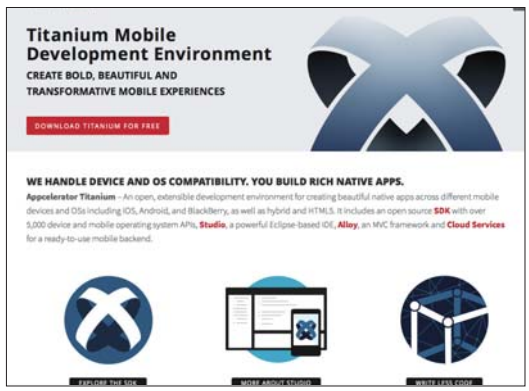


図1.3 Titanium Mobile
http://www.appcelerator.com/titanium/

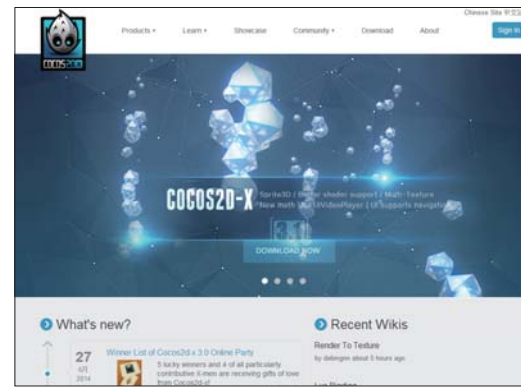


図1.4 cocos2d-x
http://www.cocos2d-x.org

図1.5 Corona SDK
http://coronalabs.com/products/corona-sdk/

図1.6 Unity
https://unity3d.com

一般アプリを作成するためのTitanium Mobileを利用しても簡単なゲームを作ることはできますが、ゲームアプリ向けのミドルウェアは、画像を表示するための機能が充実していたり、アニメーション周りの処理効率が非常に良くできる等、ゲームに特化した機能・処理が豊富に用意され、難しい処理を必要とするゲームでも簡単に作ることができます。このようにゲームアプリ向けのクロスプラットフォーム開発を行うことができ高いパフォーマンスを発揮できる主因として、OpenGL ES（OpenGL for Embedded Systems）と呼ばれる3Dコンピュータグラフィック用APIの採用があります。WindowsやMacでは一般的にOpenGLが利用されていますが、OpenGL ESはOpenGLの組み込みシステム用に開発されたものです。

アプリ用ミドルウェアの登場により、クロスプラットフォーム開発が容易になりましたが、これにより新しい問題も発生しました。リリースタイミングです。通常、GoogleのGoogle Playストアであれば、アプリのアップデートは数時間で完了しますが、AppleのApp Storeの場合Appleの審査が必要であるため、リリースできるまで1週間前後を要します。万が一リジェクトされると再度審査を受けなければいけないため、リリースはさらに伸びてしまいます。例えば、ミドルウェアだけでゲームを作成し、アプリをリリースしたとしましょう。しかし残念なことに、そのリリースしたアプリのヘルプ画面の表示内容が間違っていました。その内容を修正しなければユーザーが思い通りに遊ぶことができないため修正は急務ですが、ヘルプ画面もアプリ内に組み込んでいるので修正するにはそれぞれのストアへ申請する必要があります。しかし、App Storeで修正したアプリをリリースするには、最低でも1週間前後かかってしまいます。その間に多くのユーザーが離れてしまうでしょう。このようなことはできる限り避けたいものです。

もし、このゲームが以前のようにアプリ内ブラウザを利用していたのであれば、この問題はどうなっていたでしょう？　ヘルプ画面をサーバーに用意していたのであれば、サーバー内の内容を書き換えるだけで済みます。もちろん各ストアへの申請は不要です。このように、クロスプラットフォーム開発の最新の方法であるミドルウェアを使ったからと言っても、万能ではなく不利になる点も出てきます。そのため、基本的なシステムはミドルウェアを使用してアプリを作成し、ヘルプ画面・利用規約画面・問い合わせ画面等柔軟に変更が必要な画面は、アプリ内ブラウザを利用したり各OSのブラウザで表示することもあります。また、そのための仕組みがミドルウェアで用意されています。

また別の問題として、スマートフォン向けミドルウェアの歴史はまだまだ浅いため、プログラマー市場においても開発者が不足しているのが現状です。アプリ内ブラウザを利用した方法であれば、アプリ内の非ゲーム画面をWebデザイナやWebプログラマーに、手伝ってもらうことも可能になります。

1-2 cocos2d-xと他のゲーム向けミドルウェアの比較

本書で利用されているcocos2d-xについて紹介します。cocos2d-xはZhu Wang氏（中国）により産み出され、Chukong Technologies社（北京）により開発が進められています。このcocos2d-xは、元々cocos2d for iPhoneと呼ばれるiOS向けのフレームワークを、クロスプラットフォーム開発向けに改良されたものです。さらにcocos2d for iPhoneは、cocos2dと呼ばれるデスクトッププラットフォーム向けのクロスプラットフォーム開発フレームワークをiPhone向けのフレームワークとして作られたものでした。この一番の元となったcocos2dの開発者は、Ricardo Quesada氏（アメリカ）によるものでしたが、現在ではこのRicardo氏がcocos2d-xの開発に携わっています。現在の安定バージョンはcocos2d-x v3.1.1です（2014年6月現在）。

cocos2d-x、Corona SDK、Unityの比較概要

それでは、ゲームアプリを作るためのクロスプラットフォーム開発に対応したミドルウェアのうち、多くの企業で利用されているcocos2d-x、Corona SDK、Unityの比較を行ってみましょう。

表1.2　価格や開発環境の比較

	cocos2d-x	Corona SDK	Unity
価格（全機能の利用）	Free	$199 （1ヶ月・1開発者あたり）	157,500円 （複数の条件あり）
オープンソース	○	×	×
開発言語	C++ / Lua / JavaScript	Lua	C# / JavaScript / Boo
専用エディタ	△	△	○
専用シミュレータ	△	○	○
UIエディタ	○	△	○
公式サポート	△	○	○
日本語コミュニティ	○	○	○

表1.3　対応プラットフォームの比較

	cocos2d-x	Corona SDK	Unity
iOS	○	○	○
Android	○	○	○
Windows Phone 8	○	○	○
BlackBerry	○ (v2.x)	×	○
Marmalade	○ (v2.x)	×	×
Windows	○	×	○
Windows 8	○	×	○
Mac	○	×	○
Linux	○	×	×
Web	△ (cocos2d-JS)	×	○ (Unity Web Player)

価格

まずは開発者にとって一番気になる価格です。

cocos2d-xは、オープンソースですので全ての機能を無料で利用することができます。これは、cocos2d-xがcocos2d for iPhoneから派生したフレームワークであり、さらにその元になったcocos2dの時代からオープンソースであったことに由来します。補足すると、cocos2d-xはMITライセンスを採用しています。MITライセンスをごく簡単に説明すると、「誰でも無償で無制限に利用可能である」ライセンスです（詳細については、本節の最後に示しています）。

Corona SDKにおいても、アプリの作成からリリースまで全て無料で利用することができますが、近年主流となっているフリーミアムモデルのアプリを作成しようとするとアプリ内課金が必要になり、このアプリ内課金の利用にはライセンス料金が必要になります。またビルドを行う場合、Coronaビルドサーバーへアクセスする必要があります。オフラインビルドも可能ですが、やはりライセンス料金が必要になります。その他にもいろいろと制限がありますが、全機能を利用しようと　すると、1開発者あたり月額$199です。

Unityも、アプリの作成からリリースまで全て無料で利用することができます。しかし無料で作成したアプリには、アプリ起動時にUnityのロゴが必ず表示されてしまいます。その他、ゲームのクオリティ（シャドーや反射等）を上げようとすると、やはり無料版では制限がかかります。全ての機能を利用できるUnity Proは、157,500円と非常に高価なものになっています。もし複数人で開発する場合は、全ての開発者が同じバージョンを利用する必要があります。

開発言語

開発言語も三者三様です。

cocos2d-xは、cocos2d-x自身の開発がC++で行われているため、C++で開発できるのはもちろん、LuaやJavaScriptでも開発できます。Corona SDKは、Lua一択となります。Unityは、C#・JavaScript・Booを利用することができます。

オブジェクト指向での開発を好むのであればC++やC#が使えるミドルウェアを、Webからの転向でJavaScriptが得意なのであればJavaScriptが利用できるミドルウェアを、過去のゲーム開発でLuaに馴染み深いのであれば、Luaが利用できるミドルウェアを選択しても良いでしょう。

開発環境

開発を行っている時、利便性の高い開発環境があると、開発の効率が良くなります。

cocos2d-xは、フレームワークのみの提供となるため、専用コードエディタや専用シミュレータは持っていません。開発環境はそれぞれのOSにより異なりますが、例えばiOSアプリを基準にして開発を進めるのであればMacのXcodeを、Androidアプリを基準にして開発を進めるのであればEclipseを利用する方が多いです。そのためシミュレータは、その開発環境に依存しiOSシミュレータやAndroidエミュレータを利用することになります。また最終的にアプリの実行ファイル（iOSのipaファイル、Androidのapkファイル等）を作成する時は、各OSに応じた環境が必要になります。あまり良い状況とは言い難いですが、これも今後は大きく変わります。それについては、次節の「UIエディタ」で紹介します。

しかし今後この状況も変わるでしょう。というのは、現在Chikong Technologies社がCocos Code IDEと呼ばれるcocos2d-x用のエディタを開発しているからです。これもcocos2d-xと同じく無料で利用することが可能です。利用できる開発言語は、LuaとJavaScriptとなります。公式サイトでβ版が公開されているので、一足先に利用してみてはいかがでしょうか？（http://www.cocos2d-x.org/wiki/Code_Editor）

一方Corona SDKは、専用コードエディタではありませんがCorona Editor（図1.7）と呼ばれるSublime Text（図1.8）のプラグインが用意されています。またCorona Simulatorと呼ばれる専用シミュレータが用意されており、保存した内容を瞬時に表示することができる優れものです。

Unityはと言いますと、Unityが統合環境であるため、専用コードエディタも専用シミュレータも標準で装備されています。コードエディタは、Unityとともにインストールされるが利用できます。

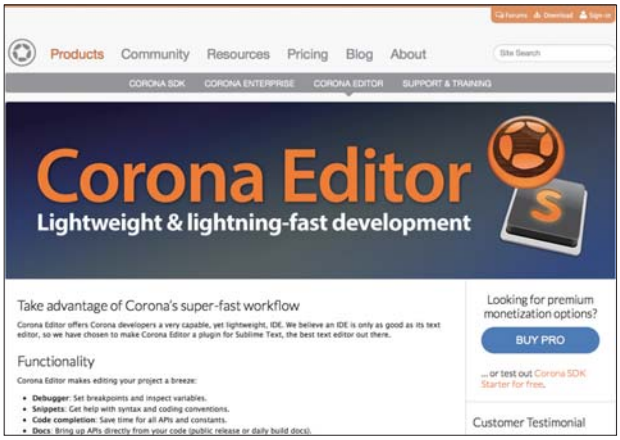


図1.7 Corona Editor
http://coronalabs.com/products/editor/

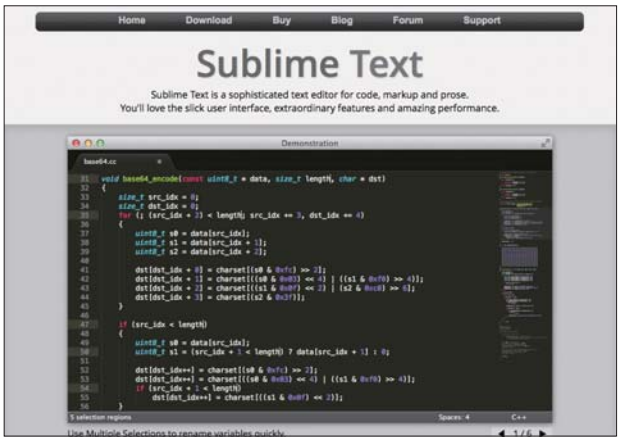


図1.8 Sublime Text
http://www.sublimetext.com

UIエディタ

ではゲーム画面を作成するためのUIエディタについてはどうでしょうか。

先の説明で、cocos2d-xはフレームワークのみを提供していると紹介しましたが、cocos2d-xの開発とともにCocoStudio（図1.9）と呼ばれるUIエディタの開発も進められています。このCocoStudioはcocos2d-x専用であり、UIエディタの他にアニメーションエディタやデータエディタ等多くの機能を有しています。

CocoStudioはcocos2d-xとともにバージョンアップを行っており、公式ツールとして安心して利用することができます。cocos2d-x用のエディタであるため、UIエディタもアニメーションエディタも2Dの画面やキャラクターを作成するために最適化されたツールですが、かなり多くの機能を有しています。UIエディタでは、画面や各パーツを作るのはもちろんのこと、パーティクルを作成することもタップイベントやサウンドを追加することも可能です。アニメーションエディタでは、ボーンアニメーション（スケルタルアニメーション）と呼ばれる技術を利用することができます。ボーンアニメーションとは、キャラクター等に対して骨格（ボーン）を定義しそれらを動かすことで、キャラクターをより自然に動かすことができるアニメーションです。またCocoStudioのホームページは中国語ですが、CocoStudio自身

図1.9 CocoStudio
http://www.cocos2d-x.org/wiki/CocoStudio

は多言語対応しており、日本語での表示も可能ですので中国語や英語を理解できなくても問題ありません。現バージョンであるCocoStudio v1.4.0.1（2014年6月現在）ではまだWindowsしか動かすことができませんが、MacやLinuxにも対応する予定であると公式に発表されています。気になる価格ですが、これだけの機能を持っていながら、なんとcocos2d-xと同じく無料で全機能を利用することができます。

またマップエディタとして良く知られているTiled Map Editor（図1.10）もサポートされており、cocos2d-xにマップエディタデータを読み込むためのクラスが用意されています。その他、サードパーティ製のエディタも複数存在していますが、頻繁に行われるcocos2d-xのバージョンアップに追いついていないのが現状です。これからcocos2d-x向けのUIエディタを利用するのであれば、公式にサポートされているCocoStudioがお勧めです。なお、CocoStudio v1.0.0.0がリリースされたのは2013年9月で、それ以前はCocosBuilder[*1]と呼ばれるUIエディタが一般的に利用されていました。現在ではこのCocosBuilderの開発は停止しています。

MEMO

***1　CocosBuilder**

開発は停止していますが、cocos2d-xの最新バージョンでもサポートは続いています。また、後継としてSpriteBuilderが開発されていますが、cocos2d for iPhoneのみの対応です。
https://github.com/cocos2d/CocosBuilder

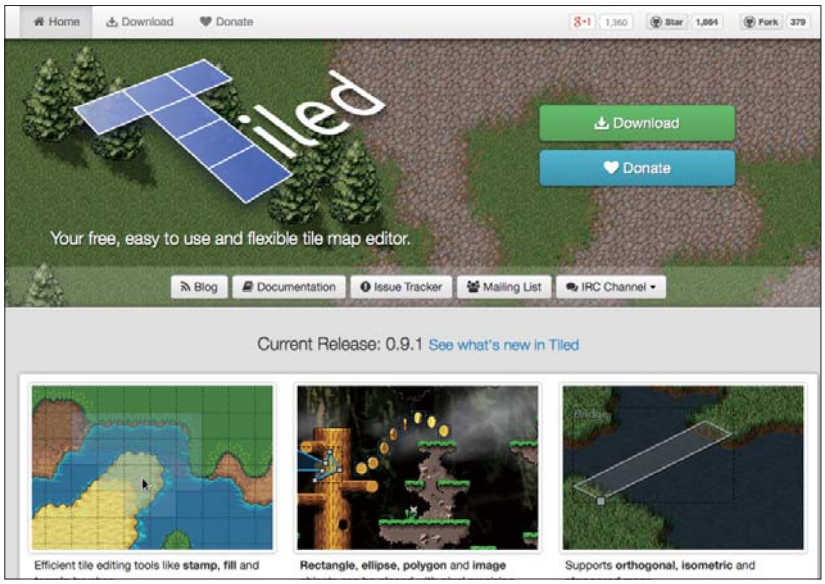


図1.10 Tiled Map Editor
http://www.mapeditor.org

Corona SDKには専用のエディタが存在しませんが、LevelHelper（図1.11）と呼ばれるエディタを利用することで、UIを作成することができます。こちらは有料ツールではありますが、それだけ多くの機能が付いているので、アプリ作成を助けてくれるでしょう。

Unityは統合環境ですので、専用のUIエディタが利用できます。また専用シミュレータとも密接に連携しているので、自由度は非常に高いと言えるでしょう。またUnityは、ユーザー開発用のアセットを販売する仕組みであるAsset Store(図1.12)を持っており、ここから豊富なエディタを追加することもできます（エディタだけでなく、多くのリソースやスクリプトもダウンロードすることも可能です）。これはUnityが成功している要因のひとつです。

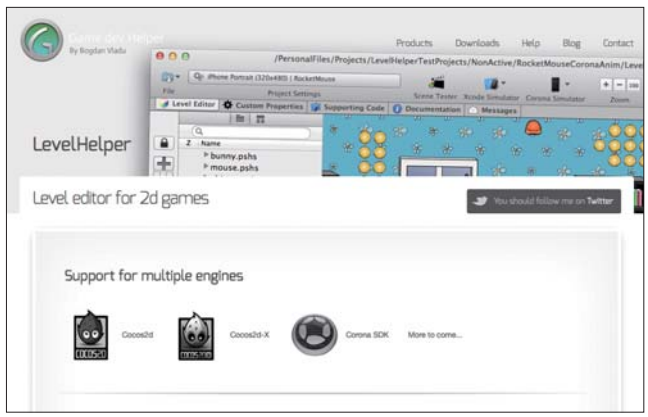


図1.11 LevelHelper
http://www.gamedevhelper.com/levelhelper/

図1.12 Unity Asset Store
https://www.assetstore.unity3d.com/en/

公式サポート・日本語コミュニティ

実際に開発が進んだ時、多くの問題に遭遇するでしょう。その時公式サポートに頼ることもありますが、それぞれのミドルウェアではどのようにサポートしてもらえるのでしょうか。

cocos2d-xは、公式ホームページにフォーラム[*1]が用意されています。このフォーラムでは毎日多くの投稿があり、cocos2d-xの開発者自身もその討論に加わっています。非常に活発ではありますが、全て英語で会話されています。英語が達者な方であれば問題ありませんが、そもそもオープンソースであり無償で利用しているため100％回答が得られるわけではありません。その辺は心得ていた方が良いでしょう。英語が苦手な方は、日本語コミュニティを頼っても良いでしょう。日本では、2013年3月に設立された日本cocos2d-xユーザー会[*2]（図1.13）があります。このコミュニティはcocos2d-x.jp community[*3]としてFacebookグループで展開されており、中にはcocos2d-xの開発元であるChukong Technologies社の日本語ができる開発者も参加しています。

Corona SDKには、公式のサポートを受けることができる仕組みがあります。ただし、PROライセンス（月額$49）を保持している開発者が受けることができるプレミアムサポートであり、5時間あたり$1125と非常に高価なものになっています。確実にサポートを得られますが、個人ユーザーにとってはあまり現実的ではないですので、その時はコミュニティを活用しましょう。日本では、日本コロナの会がGoogleグループで展開されている[*4]ほか、Corona SDK Developers Japan[*5]がFacebookグループにあります。

図1.13　日本cocos2d-xユーザー会
http://cocos2d-x.jp

Unityにおいて技術的な質問は、基本的にフォーラムやQ&Aサイトを利用することになっています。もしそれ以上のサポートが必要である場合は、有償のプレミアムサポートを受けることもできますが、フォーラム等も含め全て英語での受け答えが必要です。しかしUnityに関する日本語の情報は、比較的多いので不自由することはないでしょう。初心者レベルですと、インターネットを検索すると直に目的の回答が得られると思います。またUnityには、多くの非公式コミュニティが存在します。その中でも4000人以上のメンバーがいるUnityユーザー助け合い所は、誰でも参加することができます[*6]。

MEMO

***1　cocos2d-xフォーラム**
http://www.cocos2d-x.org/forums

***2　日本cocos2d-xユーザー会(cocos2d-x.jp)**
http://www.cocos2d-x.jp

***3　cocos2d-x.jp community**
https://www.facebook.com/groups/cocos2dxjp/

***4　日本コロナの会**
https://groups.google.com/forum/#!aboutgroup/coronaja

***5　Corona SDK Developers Japan**
https://www.facebook.com/groups/226496994147323/

***6　Unityユーザー助け合い所**
https://www.facebook.com/groups/unityuserj

対応プラットフォーム

冒頭でも紹介したように、現在はiOSとAndroidに対応するだけで比較的多くのユーザーに触れてもらうことができるでしょう。しかし、毎年新しいOSが生まれているので、いつシェアが変わるか誰もわかりません。

cocos2d-xがサポートしているモバイルプラットフォームは、iOSとAndroidのほかに、Windows Phone 8にも対応しています（v2.Xでは・BlackBerry・Marmaladeにも対応）。デスクプラットフォームではWindows・Mac・Linuxにも対応しているので、スマートフォンだけでなくPCへの展開も可能です。さらにcocos2d-xから派生したポートとしてcocos2d-JSを利用することでWebにも対応することができます。cocos2d-xでは開発言語にJavaScriptを選択できますが、同じコードを利用してcocos2d-JSでの開発が可能となっています。つまり、スマートフォン・PC・Webとオールマイティなフレームワークであることがわかるでしょ

試し読みはお楽しみ
いただけましたか？

ここからはManatee
おすすめの商品を
ご紹介します。

Tech Book Zone
Manatee

3.15
2017

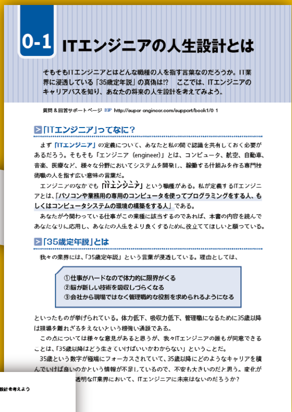
Tech Book Zone Manatee

おすすめ電子書籍

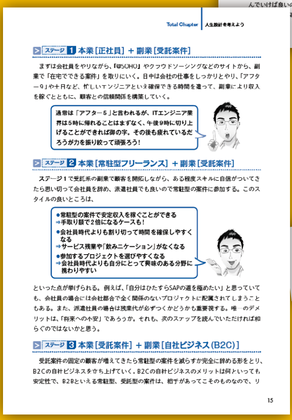
1

副業? 独立? それとも? ITエンジニアの人生設計の決定版!

組織を束ねるマネジャーになるか、現場のスペシャリストであり続けるか。自分の技術を活かして独立するか、副業を考えるか……。意外に悩ましいITエンジニアの人生設計。会社に依存しない、転職や独立も射程に入れた「マインドセット」の持ち方から、お金、営業戦略、顧客対応術、ビジネスモデルの構築といった「ビジネスロジック」まで。求人情報ポータルサイト「@SOHO」の開発者が、自身の知見と経験から得たノウハウを教えます。



トピックスごとにオンラインの公式サイトで質問コーナーを用意。疑問に答えてくれる



ITエンジニアがどのように人生を描き切り開いていけるか、著者の実体験をもとに解説

IT エンジニアのための「人生戦略」の教科書
技術を武器に、充実した人生を送るための「ビジネス」と「マインドセット」

マイナビ出版
平城寿（著者） 256 ページ
価格：2,462 円（PDF・EPUB）

開発系読み物

「開発技法」

2

UE4におけるゲーム制作で必須の知識と経験が身につく!

『Unreal Engine 4で極めるゲーム開発』は、Unreal Engine 4（UE4）の機能を単に紹介するだけでなく、一本のサンプルゲーム開発に順序よく盛り付けていく構成になっており、機能の組み合わせ方や、実践的な使い方をラーニングできます。3Dゲーム開発の一般的なワークフローやプロセスも解説し、章の構成も実際の開発プロセスに近づけました。今使っている人にも、これから始める人にも、すべてのUE4ユーザーにオススメの一冊!

推奨職種を併記。アーティストやレベルデザイナなど専門ごとに読むべき章がわかる

プログラムやスクリプトの勉強をしたことがない人でも読み進められる

Unreal Engine 4 で極めるゲーム開発

ボーンデジタル
湊和久（著者）
592 ページ
価格：4,860 円（PDF）

開発ツール

ワークフローを疑似体験！ GitHub が初歩からわかる

開発ツール

3

Web 制作者のための GitHub の教科書 チームの効率を最大化する 共同開発ツール

Web 制作における「GitHub」の使い方が、実際のワークフローをイメージしながら理解できます。「そもそもどんなサービスなの ?」「どういうときにどの機能を使えばいいの ?」といった初歩の疑問から解説します。

インプレス
塩谷啓・紫竹佑騎・原一成・平木聡（著者）
224 ページ　価格：2,052 円（PDF）

&

Docker が利用される 現場のノウハウが凝縮！

開発ツール

4

Docker 実践ガイド

Docker が利用される環境や背景をはじめ、導入前のシステム設計、基本的な利用方法、Dockerfile による自動化の手法、プロセッサ、ネットワーク、ストレージなどの資源管理、管理・監視ツールについて解説します。

インプレス
古賀政純（著者）
328 ページ　価格：3,240 円（PDF）

&

チーム改善に活かす ITIL 悩めるリーダーにオススメ

開発系読み物

5

新米主任 ITIL 使ってチーム改善します！

化粧品メーカーで主任に昇格した友原京子。異動先は問題だらけのハチャメチャ部署だった…。『新人ガール ITIL 使って業務プロセス改善します !』の第 2 弾。英国生まれの IT 運用ノウハウ「ITIL」をチーム改善に活かします。

シーアンドアール研究所
沢渡あまね（著者）
304 ページ　価格：1,750 円（PDF）

プロトタイピングによって 初期段階での可能性を探る

開発系読み物

6

プロトタイピング実践ガイド スマホアプリの効率的なデザイン手法

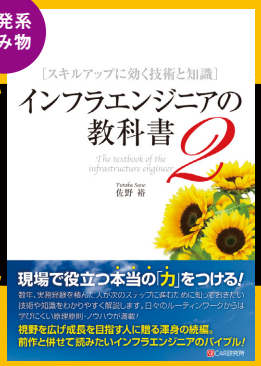
本書で解説するプロトタイピングは、紙などを使った「低精度プロトタイピング」を中心とした手法です。設計フェーズの早期段階から作成し、検証と改善によって、機能要件や UI 設計、デザインを具現化していけます。

インプレス
深津貴之・荻野博章（著者）
240 ページ　価格：2,592 円（PDF）

&

インフラエンジニアの 必須知識をていねいに解説

開発系読み物



7

インフラエンジニアの教科書 2 スキルアップに効く技術と知識

数年間インフラエンジニアの経験を積んでいても「自分は詳しく知らないし、他の人に説明できない」といったことがあります。本書は実務経験を積んだインフラエンジニアを対象に、必須知識をわかりやすく解説します。

シーアンドアール研究所
佐野裕（著者）　価格：2,070 円（PDF・EPUB）

&

エミュレータ制作を通して コンピュータの中身を理解

開発系読み物

8

自作エミュレータで学ぶ x86 アーキテクチャ コンピュータが動く仕組みを徹底理解！

機械語やアセンブリ言語が CPU でどう実行されるか意識することはめったにありません。本書ではエミュレータの制作を通して x86 CPU の仕組み、メモリ・キーボード・ディスプレイといった部品と CPU の関わりを学びます。

マイナビ出版　内田公太・上川大介（著者）　196 ページ
価格：2,324 円（PDF）