

Roland Willms/Yann Heeser

Know-how
ist blau.



Spiele entwickeln für iOS und Android mit Cocos2D

- > Programmieren für die großen Smartphone-Plattformen
- > Entwickeln mit einem genialen Framework
- > Von der Spielidee bis zum App-Store

FRANZIS

Roland Willms/Yann Heeser

**Spiele entwickeln für iOS
und Android mit Cocos2D**

Roland Willms/Yann Heeser

Spiele entwickeln für iOS und Android mit Cocos2D

Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

Alle Angaben in diesem Buch wurden vom Autor mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. Der Verlag und der Autor sehen sich deshalb gezwungen, darauf hinzuweisen, dass sie weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen können. Für die Mitteilung etwaiger Fehler sind Verlag und Autor jederzeit dankbar. Internetadressen oder Versionsnummern stellen den bei Redaktionsschluss verfügbaren Informationsstand dar. Verlag und Autor übernehmen keinerlei Verantwortung oder Haftung für Veränderungen, die sich aus nicht von ihnen zu vertretenden Umständen ergeben. Evtl. beigefügte oder zum Download angebotene Dateien und Informationen dienen ausschließlich der nicht gewerblichen Nutzung. Eine gewerbliche Nutzung ist nur mit Zustimmung des Lizenzinhabers möglich.

© 2012 Franzis Verlag GmbH, 85540 Haar bei München

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Das Erstellen und Verbreiten von Kopien auf Papier, auf Datenträgern oder im Internet, insbesondere als PDF, ist nur mit ausdrücklicher Genehmigung des Verlags gestattet und wird widrigenfalls strafrechtlich verfolgt.

Die meisten Produktbezeichnungen von Hard- und Software sowie Firmennamen und Firmenlogos, die in diesem Werk genannt werden, sind in der Regel gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden. Der Verlag folgt bei den Produktbezeichnungen im Wesentlichen den Schreibweisen der Hersteller.

Lektorat: Anton Schmid

Satz: DTP-Satz A. Kugge, München

art & design: www.ideehoch2.de

Druck: C.H. Beck, Nördlingen

Printed in Germany

ISBN 978-3-645-60156-6

Vorwort

Cocos2D und Cocos2D-X

Cocos2D für iPhone ist ein Framework mit der Adresse

<http://www.cocos2d-iphone.org/>

Es dient zur Programmierung von 2-D-Spielen, ist aber auch für andere grafische und interaktive Applikationen nützlich. Es basiert auf dem Cocos2D-Design mit der Adresse

<http://www.cocos2d.org/>

und benutzt die gleichen Konzepte, ist aber in der Programmiersprache Objective C geschrieben. Neben den iOS-Geräten iPod, iPhone und iPad unterstützt Cocos2D für iPhone auch das Betriebssystem OS X.



Bild 0.1: Das Logo von Cocos2D

Cocos2D-X mit der Adresse

<http://www.cocos2d-x.org/>

ist eine Portierung von Cocos2D für iPhone in der Programmiersprache C++, um weitere Plattformen zu erschließen. Hierzu gehören mobile Systeme, zum Beispiel Android und Windows Phone, sowie Desktopsysteme, zum Beispiel Linux und Windows.



Bild 0.2: Das Logo von Cocos2D-X

Klassenspektrum von Cocos2D

Cocos2D ist einfach zu verwenden, integriert Praktiken von OpenGL ES unter iOS und Open GL unter OS X zur Optimierung der Schnelligkeit, hat eine aktive Benutzergruppe mit Forum, ist Open Source und kostenlos für eigene Spiele einsetzbar. Mehrere Tausend Spiele verwenden bereits das Cocos2D-API, darunter auch viele Bestseller.

Eine Liste verfügbarer Spiele finden Sie im Internet unter der Adresse:

<http://www.cocos2d-iphone.org/games/>

Cocos2D bietet zahlreiche Klassen für verschiedene Zwecke an: Szenenmanagement, Übergänge zwischen Szenen, Sprites, Effekte, Aktionen, Menüs, Schaltflächen, integrierte physikalische Engines, Teilchensysteme, Textunterstützung, Textur Atlas und Tile Maps, Soundmaschine, punktebasiertes Koordinatensystem zur Nutzung unterschiedlich großer Bildschirme einschließlich Retina-Bildschirme, Eingabe per Finger unter iOS und Tastatur unter OS X, Beschleunigungssensor, Hoch- und Querformat unter iOS, integrierte Pausen- und Fortsetzungsfunktion.

Am Beispiel des Projekts *Euro Crisis* beschreibt dieses Buch schrittweise die Entwicklung eines Spiels von einzelnen Elementen bis hin zum fertigen Programm. Hierbei kommen die wichtigsten Klassen von Cocos2D zum Einsatz.



Bild 0.3: Das Startmenü von Euro Crisis

Kapitel 1 befasst sich mit der Installation der nötigen Software für Cocos2D und Cocos2D-X, dem Start des ersten Programms und dem Überspielen der fertigen Software in die Stores. In Kapitel 2 erfahren Sie einige Grundlagen zum Spieldesign anhand konkreter Beispiele, um die Entwicklung eigener Ideen anzuregen. Von Kapitel 3 bis Kapitel 8 geht es um einzelne Elemente in Spielen: Bilder, Aktionen, Ereignisse, Sound, Beschriftungen und Teilchensysteme. In Kapitel 9 bekommen Sie einen Einblick in die innere Gestaltung von Spielleveln und die Koordination unterschiedlicher Stränge.



Bild 0.4: Koordination verschiedener Spielereignisse

Kapitel 10 und 11 behandeln den Rahmen um eine Spielhandlung. Hierzu gehören Szenen und Übergänge sowie Menüs und Schaltflächen zur Steuerung. In Kapitel 12 lernen Sie, wie einfach es ist, Spielereignisse und Bestenlisten lokal zu verwalten und auch global verfügbare Boards wie zum Beispiel das Game Center von Apple zu nutzen.

Objective C und C++

Obwohl Cocos2D-X nicht in der hauseigenen Sprache Objective C von Apple geschrieben ist, sind Vorbereitungen getroffen worden, dass die Programme auch unter iOS laufen. Daher müsste man eigentlich nur C++ beherrschen, um alle Plattformen mit Apps bedienen zu können.

Cocos2D für iPhone ist sehr früh auf den Markt gekommen, sodass Sie im Internet auf Websites und in Diskussionsforen meistens Quellcode in Objective C finden. Als das Framework immer häufiger in erfolgreichen Spielen eingesetzt wurde, begannen einige Entwickler damit, die Klassen auch in andere Sprachen zu portieren. Dies ist nicht einfach, weil jede Plattform spezifische Eigenschaften hat und die jeweilige grafische Umgebung berücksichtigt werden muss. So gab es für Android zunächst einen Port in Java und für Windows Phone einen Port in C#. Weil sich alle Plattformen jedoch technisch weiterentwickeln, blieb unklar, ob diese Ports in einigen Jahren noch laufen.

Mit der Übernahme der Entwickler von Cocos2D für iPhone in die Spielefirma Zynga ist sichergestellt, dass das Framework weiterhin gepflegt und an den neuesten Stand der Technik angepasst wird. Cocos2D-X ist mittlerweile relativ breit aufgestellt und wird sehr zeitnah nach Updates von Cocos2D für iPhone angepasst, sodass Sie davon ausgehen können, dass es in einigen Jahren noch laufen wird.

In diesem Buch finden Sie den Quellcode stets in zwei Varianten: zunächst in einer kommentierten Form in Objective C und immer direkt im Anschluss in unkommentierter Form in C++. So ist sichergestellt, dass Sie später beim Herumsuchen im Internet flexibel sind und Code in Objective C und C++ parallel verstehen.

Support

Das Spiel *Euro Crisis* als Programmierprojekt in diesem Buch und die Variante *Dollar Crisis* finden Sie in den Stores der jeweiligen Systeme.

Die Dateien für die Projekte in den einzelnen Kapiteln gibt es im Internet auf der Franzis-Seite:

<http://www.buch.cd/>

Hier finden Sie eine Anleitung, wie Sie mithilfe der ISBN dieses Buchs den Quellcode der Programme einschließlich der zugehörigen Ressourcen der Projekte erhalten.

Aus lizenz- und urheberrechtlichen Gründen sind die originalen Medien des Spiels wie zum Beispiel Icons, Bilder und Sounds in den angebotenen Dateien besonders gekennzeichnet oder ersetzt. Die mitgelieferten Medien dienen nur als Anschauungsmaterial im Rahmen dieses Buchs und dürfen nicht in eigenen Spielen verwendet werden. Im Internet stehen Ihnen zahlreiche Websites zur Verfügung, die Millionen Bilder und Sounds zur Lizenzierung für kommerzielle Zwecke anbieten und die Urheber entsprechend vergüten.

Für Diskussionen rund um das gesamte Klassenspektrum von Cocos2D gibt es das Forum mit der Adresse:

<http://www.cocos2d-iphone.org/forum/>

Es ist sehr gut besucht, sodass Benutzer rasch Antworten auf alltägliche Probleme bekommen. Mithilfe des Suchsystems können Sie leicht nachsehen, ob es bereits Antworten auf bestimmte Fragen oder Abhilfe bei aufgetauchten Fehlern rund um Cocos2D gibt.

Wir wünschen Ihnen viel Spaß beim Lesen des Buchs und der Programmierung Ihrer ersten Cocos2D-Apps sowie viel Erfolg bei der Vermarktung Ihrer Apps in den Stores.

Krefeld, im September 2012

Roland Willms

Yann Heeser

<http://www.cocos2d.de/>

<http://www.cocos2dx.de/>

Inhaltsverzeichnis

1	Hello Cocos2D	15
1.1	Xcode und Cocos2D installieren	15
1.1.1	Xcode installieren	15
1.1.2	Cocos2D installieren	16
1.2	Eine App für iOS erstellen	17
1.2.1	Eine App bei iTunes Connect einrichten	18
1.2.2	Eine App in Xcode anlegen	22
1.2.3	Produktbezogene Angaben ändern	24
1.2.4	Icons in eine App integrieren	25
1.2.5	Startbilder in eine App integrieren	27
1.2.6	Eine App testen	31
1.2.7	Eine App in den App Store hochladen	34
1.3	Eclipse und Cocos2D-X installieren	37
1.3.1	Eclipse installieren	37
1.3.2	Android integrieren	38
1.3.3	Cocos2D-X installieren	40
1.4	Eine App für Android erstellen	40
2	Spiele designen	45
2.1	Spiele kritisieren	45
2.1.1	Merkmale von Spielen	45
2.1.2	Zynga Slots	46
2.1.3	Tiny Wings	51
2.1.4	Fruit Ninja	55
2.2	Euro Crisis LT als Projekt	57
3	Bilder anzeigen	63
3.1	Spielszenen vorbereiten	63
3.1.1	Eine Szene mit einer Ebene bereitstellen	63
3.1.2	iPhone und iPad unterscheiden	69
3.1.3	Anpassungen für Android durchführen	70
3.2	CCNode als Hauptelement verstehen	72
3.2.1	Zustand von CCNode	73
3.2.2	Zustand von CGPoint	74
3.2.3	Verhalten von CCNode	75

3.3	Bilder mit CCSprite anzeigen	75
3.4	Bilder transformieren	77
4	Aktionen starten	83
4.1	Aktionstypen überblicken.....	83
4.1.1	Verhalten von CCAction	83
4.1.2	Abgeleitete Klassen von CCAction verwenden	83
4.2	Aktionen ablaufen lassen	89
4.3	Mehrere Aktionen verknüpfen	90
4.4	Aktionen zeitlich steuern.....	92
5	Ereignisse verarbeiten	95
5.1	Ereignisverarbeitung aktivieren.....	95
5.2	Ereignistypen unterscheiden	96
5.3	Auf Ereignisse reagieren.....	97
5.3.1	Auf Bewegungsereignisse reagieren.....	98
5.3.2	Auf Berührungseignisse reagieren	102
6	Musik abspielen	109
6.1	Den Soundplayer starten.....	109
6.2	Benutzereinstellungen respektieren.....	110
6.3	Sound abspielen	112
7	Ebenen beschriften.....	115
7.1	Etiketten anzeigen	115
7.2	Etiketten mit eigenen Zeichen gestalten.....	119
7.2.1	Zeichen mit dem Glyph Designer gestalten.....	119
7.2.2	Eigene Zeichensätze verwenden	120
7.3	Stränge einführen	122
8	Teilchensysteme erzeugen	127
8.1	Teilchensysteme mit eigenen Bildern gestalten.....	127
8.2	Teilchensysteme starten und stoppen	135
8.3	Teilchensysteme selbst programmieren	139
9	Spiellevel gestalten	149
9.1	Eigenschaften eines Levels festlegen	149
9.2	Einen Level initialisieren	153
9.2.1	Eigenschaften initialisieren.....	153
9.2.2	Eigenschaften deallozieren.....	161
9.3	Benutzereingaben verarbeiten	162
9.3.1	Den Officer bewegen.....	162

9.3.2	Einen Schuss abgeben.....	164
9.3.3	Ein Spiel beenden.....	168
9.4	Spielelemente zufällig erscheinen lassen.....	171
9.4.1	Leute hinzufügen.....	171
9.4.2	Freunde hinzufügen.....	172
9.4.3	Feinde hinzufügen.....	175
9.4.4	Kugeln der Feinde abfeuern.....	179
9.4.5	Banknoten hinzufügen.....	182
9.5	Den Spielablauf koordinieren.....	186
9.6	Ein Achievement melden.....	208
10	Szenen verwalten.....	219
10.1	Szenen mit Ebenen aufbauen.....	219
10.2	Eine Szene starten oder ersetzen.....	222
10.3	Übergänge zwischen zwei Szenen.....	223
11	Menüs einbauen.....	229
11.1	Schaltflächen vorsehen.....	229
11.2	Optionen auswählen.....	241
12	Bestenlisten speichern.....	249
12.1	Spielerdaten lokal speichern.....	249
12.2	Mit dem Game Center umgehen.....	267
12.2.1	Einen Spieler authentifizieren.....	267
12.2.2	Bestenlisten und Erfolge hochladen.....	268
12.2.3	Bestenlisten und Erfolge anzeigen.....	269
12.2.4	Bestenlisten und Erfolge im Game Center einrichten.....	273
12.3	Anzeigen für Spielerdaten einbauen.....	279
	Stichwortverzeichnis.....	285

1 Hello Cocos2D

In diesem Kapitel behandeln wir:

- die Installation von Xcode und Cocos2D zur Programmierung einer App für iOS und OS X
- die Einrichtung einer App bei iTunes Connect mit den nötigen Informationen für den App Store
- das Anlegen eines Projekts für eine App in Xcode
- die Einstellung produktbezogener Angaben und die Integration eigener Icons und Startbilder in Xcode
- die Ausführung einer App in Simulatoren und auf angeschlossenen Geräten
- das Hochladen einer App in den App Store zur Verifizierung
- die Installation von Eclipse, dem Android SDK mit NDK und Cocos2D-X zur Programmierung einer App für Android
- das Erstellen eines Android-Projekts mit anschließendem Importieren in Eclipse
- das Hinzufügen eines eigenen Hintergrundbildes und die Integration eigener Icons
- die Ausführung einer App im Android-Simulator und auf Android-Geräten
- das Hochladen einer App in verschiedene Stores

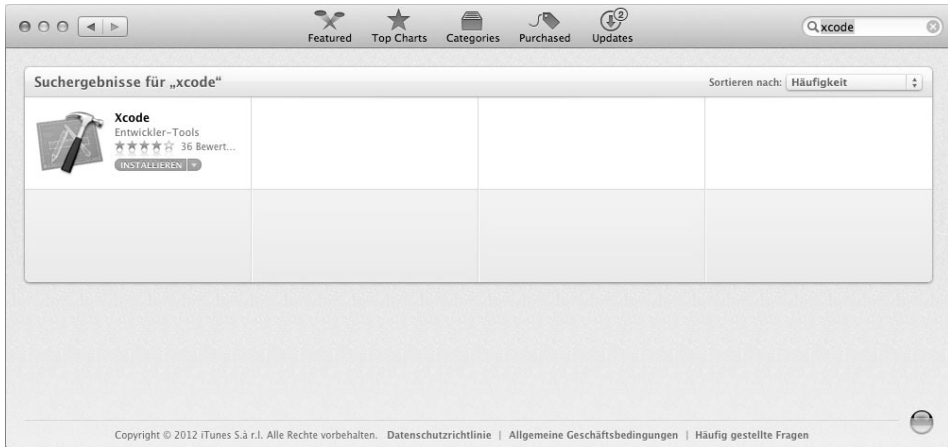
1.1 Xcode und Cocos2D installieren

Zur Entwicklung von Apps unter iOS und OS X mithilfe von Cocos2D benötigen Sie die Entwicklungsumgebung Xcode von Apple und das Cocos2D-Framework.

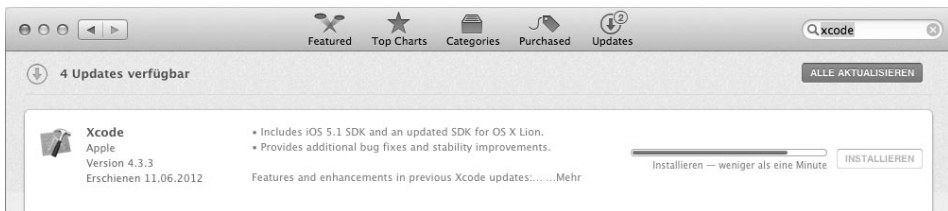
1.1.1 Xcode installieren

Xcode ist eine Software von Apple zur Entwicklung von Programmen für die Betriebssysteme iOS (iPod, iPhone, iPad) und OS X (iMac, MacBook).

Starten Sie den App Store, suchen Sie nach »xcode« und installieren Sie die Software.

**Bild 1.1:** Xcode im App Store

Spätere Updates von Xcode können Sie jederzeit wieder über den App Store installieren.

**Bild 1.2:** Updates im App Store

1.1.2 Cocos2D installieren

Besuchen Sie die Homepage von Cocos2D:

<http://www.cocos2d-iphone.org/>

Am 09.07.2012 ist nach mehr als einjähriger Entwicklung die Version 2.0 des Frameworks zur Entwicklung von 2-D-Spielen erschienen.

Im Download-Bereich finden Sie das komprimierte Archiv

cocos2d-iphone-2.0.tar.gz

mit der Cocos2D-Software. Es enthält einen Ordner mit den Dateien von Cocos2D. Kopieren Sie ihn auf den Desktop des Macs, und geben Sie ihm zur Vereinfachung den Namen:

cocos2d

Der Ordner enthält das Installationsskript:

install-templates.sh

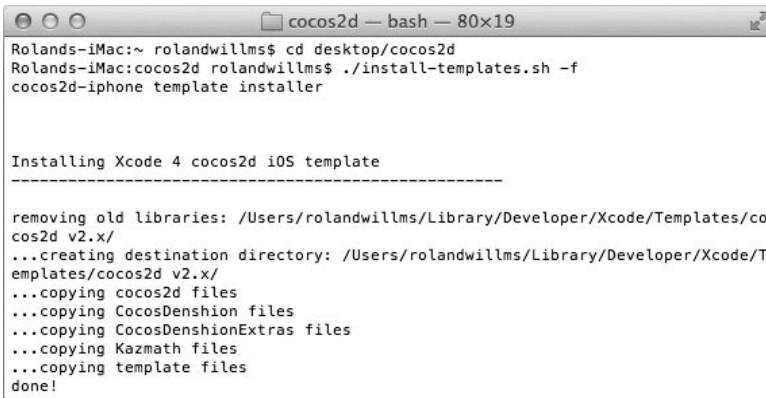
Starten Sie *Finder/Programme/Dienstprogramme/Terminal* und geben Sie den Befehl

```
cd desktop/cocos2d
```

zum Wechseln in den Ordner *cocos2d* auf dem Desktop und den Befehl

```
./install-templates.sh -f
```

zum Installieren des Cocos2D-Frameworks ein.

A screenshot of a terminal window titled 'cocos2d — bash — 80x19'. The terminal shows the following commands and output:
Rolands-iMac:~ rolandwillms\$ cd desktop/cocos2d
Rolands-iMac:cocos2d rolandwillms\$./install-templates.sh -f
cocos2d-iphone template installer

Installing Xcode 4 cocos2d iOS template

removing old libraries: /Users/rolandwillms/Library/Developer/Xcode/Templates/cocos2d v2.x/
...creating destination directory: /Users/rolandwillms/Library/Developer/Xcode/Templates/cocos2d v2.x/
...copying cocos2d files
...copying CocosDenshion files
...copying CocosDenshionExtras files
...copying Kazmath files
...copying template files
done!

Bild 1.3: Installationsskript von Cocos2D

Das Skript kopiert die Dateien von Cocos2D in die Ordner

Library/Developer/Xcode/Templates/cocos2d v2.x

und

Library/Developer/Xcode/Templates/File Templates/cocos2d v2.x

1.2 Eine App für iOS erstellen

Auf der Homepage

<https://developer.apple.com/>

finden Sie Informationen von Apple zum iOS Developer Program (iPod, iPhone und iPad) und Mac Developer Program (iMac, MacBook).

Sie müssen Mitglied bei einem solchen Programm sein, um die entwickelten Apps auf realen Geräten testen und anschließend in den App Store hochladen zu können. Ohne eine solche Mitgliedschaft ist es nur möglich, Apps in den Simulatoren zu testen.

Im iOS Provisioning Portal finden Sie eine ausführliche Anleitung zur Installation eines iOS Development Certificate. Xcode benötigt ein solches Zertifikat, um den Test auf Geräten und das Hochladen in den App Store zu ermöglichen.

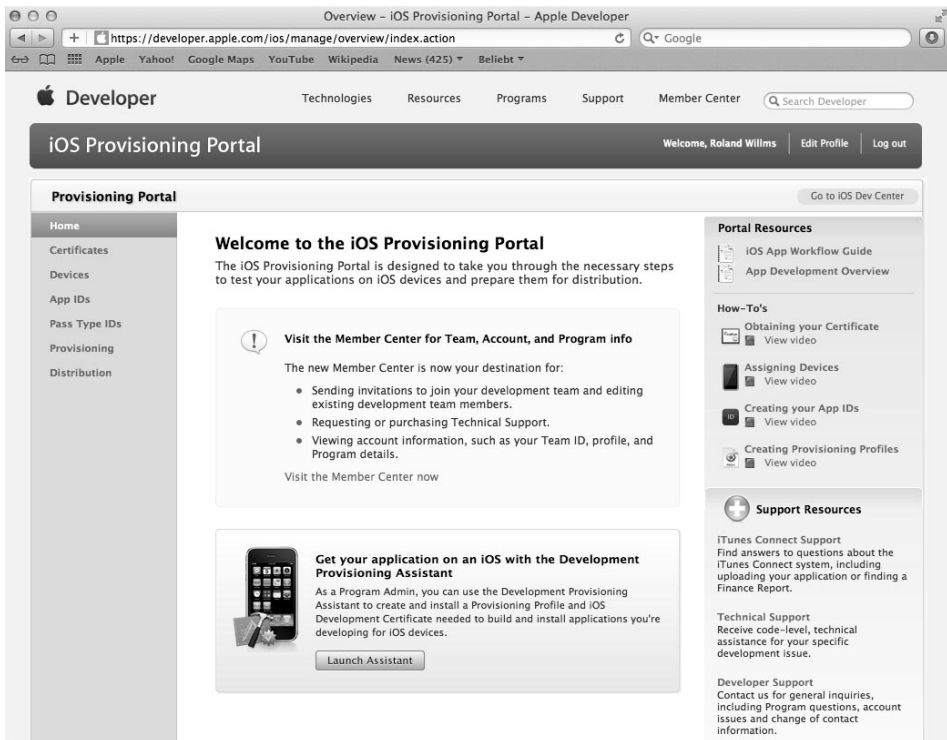


Bild 1.4: Informationen im iOS Provisioning Portal

Im Folgenden wird angenommen, dass Sie eine Mitgliedschaft besitzen und das Zertifikat zur Entwicklung von iOS Apps in Xcode installiert haben.

1.2.1 Eine App bei iTunes Connect einrichten

Loggen Sie sich bei iTunes Connect auf der Homepage

<https://itunesconnect.apple.com/>

ein. In Ihrem persönlichen Bereich finden Sie wichtige Links zu unterschiedlichen Themen.

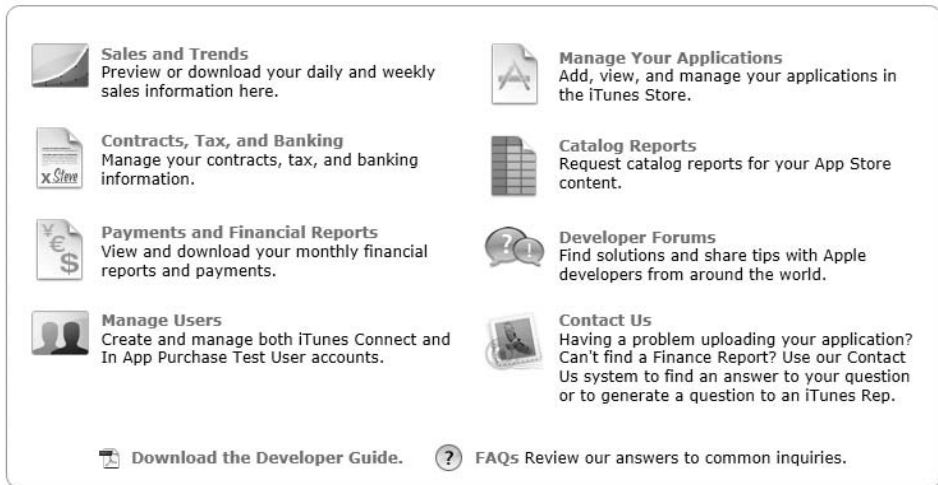


Bild 1.5: Wichtige Links in iTunes Connect

Sie sollten sich den angebotenen Developer Guide unbedingt herunterladen. Er enthält alle Informationen zu den Verträgen, zur Einrichtung, zum Hochladen und zur Verwaltung von Apps, zum Nachkaufen von Dingen mithilfe von In-App Purchases, zur Einrichtung des Game Centers, zur Verwaltung von Benutzerdaten in der iCloud und zur Werbung mithilfe des iAd-Netzwerks.

Um eine App einzurichten, folgen Sie dem Link *Manage your Applications*. Auf der nächsten Seite nutzen Sie die Schaltfläche *Add New App* und wählen auf der nächsten Seite den Typ *iOS App* aus.

Alle Angaben, die auf den folgenden Seiten gemacht werden, lassen sich später noch ändern, bis auf den App-Namen und die SKU-Nummer. Zunächst können Sie die Angaben grob eintragen. Bevor die App in den App Store hochgeladen wird, sollten aber alle Angaben vollständig ergänzt sein.

Auf der nächsten Seite geben Sie eine Sprache, einen App-Namen, eine SKU-Nummer und eine Bundle ID an. Der App-Name bleibt 180 Tage lang reserviert. Wenn nach dem Ablauf keine App hochgeladen ist, wird der Name für immer für den Account gesperrt und danach für andere Accounts freigegeben. Um den Namen zu schützen, können Sie einen Dummy als App hochladen und anschließend sofort wieder löschen.

App Information

Enter the following information about your app.

Default Language: English ⓘ

App Name: Euro Crisis LT ⓘ

SKU Number: EUROCRISISLT ⓘ

Bundle ID: Euro Crisis LT - de.cocos2d.eurocrisislt ⓘ

You can register a new Bundle ID here.

⚠ Note that the Bundle ID cannot be changed if the first version of your app has been approved or if you have enabled Game Center or the iAd Network.

Does your app have specific device requirements? [Learn more](#)

Cancel Continue

Bild 1.6: App Informationen

Die Bundle ID müssen Sie vorher in Ihrem Developer-Account angelegt haben. Wenn Sie auf den Link [here](#) klicken, gelangen Sie automatisch an die richtige Stelle und können dies nachholen.

Mithilfe der Bundle ID wird die App später weltweit identifiziert. Sie ist außerordentlich wichtig und sollte den allgemeinen Aufbau

```
<website>.<name>
```

haben, damit sie weltweit eindeutig ist.

Auf der nächsten Seite geben Sie das Verfügbarkeitsdatum an, ab dem die App im Store sichtbar sein soll, und stellen den gewünschten Preis ein.

Euro Crisis LT

Select the availability date and price tier for your app.

Availability Date: 09/Sep ⓘ 1 ⓘ 2012 ⓘ

Price Tier: Free ⓘ

[View Pricing Matrix](#)

Discount for Educational Institutions ☐ ⓘ

Custom B2B App ☐ ⓘ

Unless you select specific stores, your app will be for sale in all App Stores worldwide.

Go Back Continue

Bild 1.7: Verfügbarkeitsdatum und Preis

Auf der nächsten Seite geben Sie noch einige Dinge an, zum Beispiel die Versionsnummer, die Beschreibung im Store und die Altersbeschränkung. Auch Icons und Screenshots werden hochgeladen, vorerst am besten weiße Bilder.

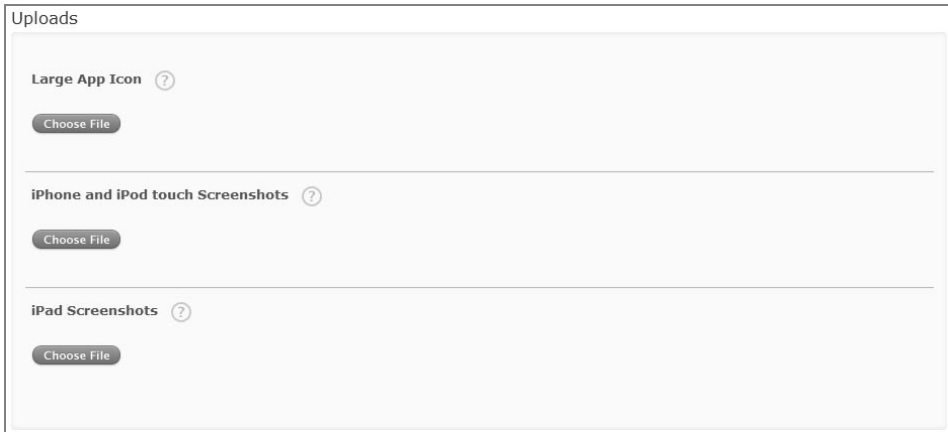


Bild 1.8: Icons und Screenshots

Die folgende Tabelle zeigt die Größen der einzelnen Bilder.

Bildtyp	Größe
Large App Icon	512 x 512 oder 1024 x 1024 (Retina)
iPhone / iPod Screenshots	960 x 640 oder 640 x 960 (Hoch- oder Querformat, beides Retina)
iPad Screenshots	1024 x 768, 768 x 1024 (Hoch- oder Querformat) oder 2048 x 1536, 1536 x 2048 (Hoch- oder Querformat, beides Retina)

Es empfiehlt sich, grundsätzlich nur noch Retina-Screenshots hochzuladen, um den perfekten Look auf den zugehörigen Displays zu bekommen. Andernfalls sehen die Informationen im Store auf den zunehmenden Retina-Geräten in einigen Jahren automatisch veraltet aus.

Nach der Bestätigung aller Angaben ist die App in iTunes Connect eingerichtet. Sie befindet sich nun im Status *Prepare for Upload*.

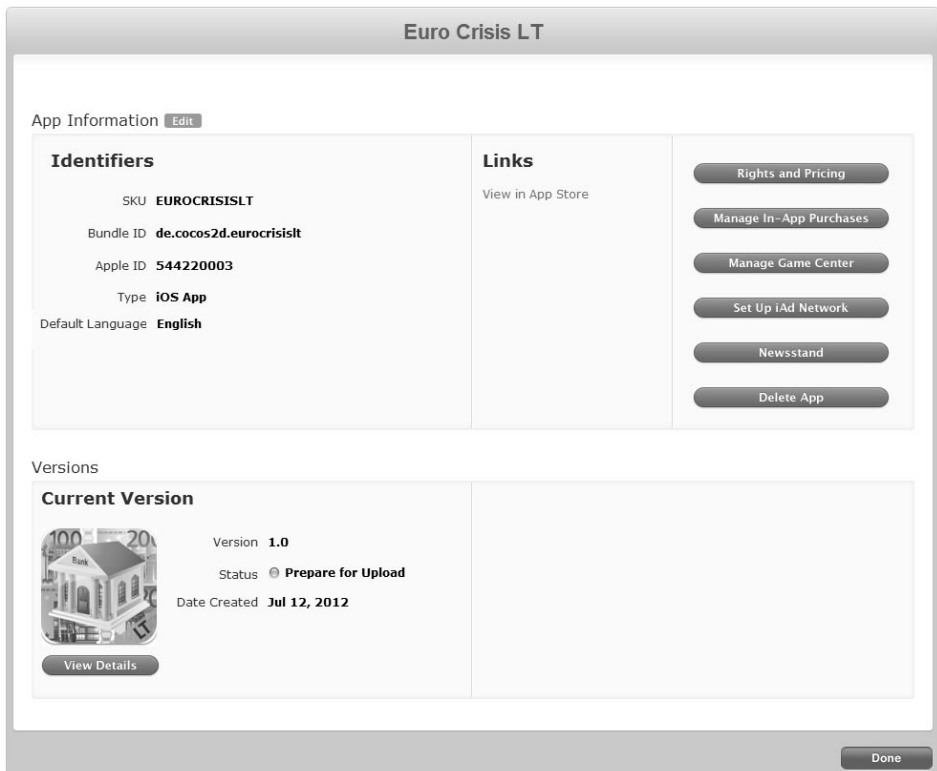


Bild 1.9: Informationen zur App in iTunes Connect

1.2.2 Eine App in Xcode anlegen

Starten Sie *Finder/Programme/Xcode* und rufen Sie das Menü *File/New/Project* auf. Unter iOS und Mac OS X befindet sich jeweils ein Eintrag *cocos2d v2.x* für das Cocos2D-Framework. Wenn Sie diesen Eintrag bei iOS auswählen, erscheinen drei Templates.

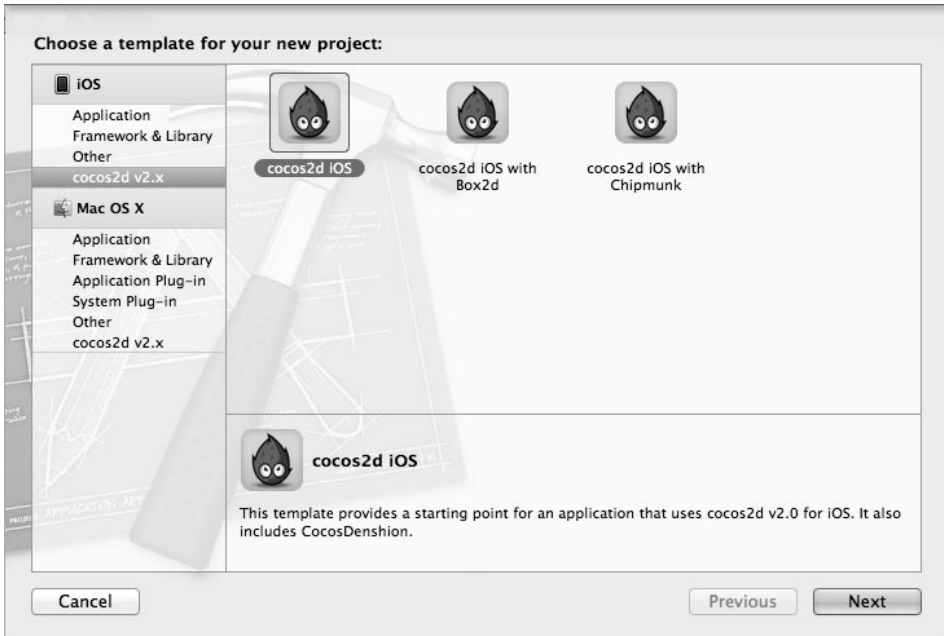


Bild 1.10: Auswahl von Cocos2D-Templates

Box2D und Chipmunk sind zwei sehr erfolgreiche Sets zur Entwicklung von Spielen, die mit physikalischen Eigenschaften wie zum Beispiel Schwerkraft und Mehrteilchensystemen arbeiten. Die Beschreibung dieser Sets ist im Rahmen dieses einführenden Buchs in Cocos2D nicht möglich.

Wenn Sie das erste Template *cocos2d iOS* auswählen, erscheint ein Dialog mit einigen Fragen.

Am wichtigsten ist die Angabe bei *Product Name*. Beachten Sie, dass dieser als Ordner- und Dateiname des Projekts verwendet wird. Insofern sollten Sie auf Leerzeichen und Umlaute verzichten. Im Screenshot ist *eurocrisislt* angegeben.

Bei *Company Identifier* sollte eine Website in umgekehrter Reihenfolge, also z. B. mit vorangestelltem *.de.* stehen, die auch tatsächlich registriert ist, um weltweite Überschneidungen zu vermeiden.

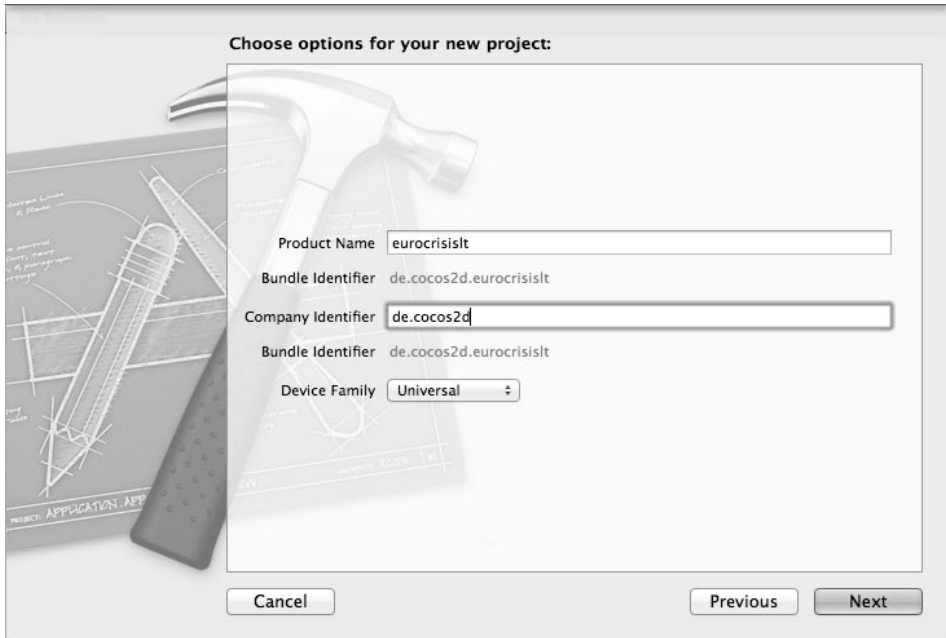


Bild 1.11: Dialog bei Auswahl von *cocos2d OS*

Der *Bundle Identifier* entsteht zwar automatisch aus dem Produktnamen und dem Firmennamen, kann aber später geändert werden.

Bei *Device Family* stellen Sie ein, ob die App unter iPod/iPhone oder iPad oder bei Universal sogar unter beiden Geräten gleichzeitig laufen soll. Viele Firmen produzieren bei einem Spiel mittlerweile eine LT-Version (Lite-Version) zum Testen auf iPod/iPhone/iPad, eine Version für iPod/iPhone und eine HD-Version (High Density) für iPad. In der Namensgebung wird dies deutlich, zum Beispiel *Euro Crisis LT*, *Euro Crisis* und *Euro Crisis HD*.

Nach der Bestätigung der Angaben werden Sie noch nach einem Speicherort für das Projekt gefragt. Anschließend erscheint ein fertig programmiertes Standardprogramm in Xcode mit eingebettetem Cocos2D-Framework. In diesem Quellcode müssen Sie nun an einigen Stellen Änderungen vornehmen.

1.2.3 Produktbezogene Angaben ändern

Xcode ist eine Entwicklungsumgebung mit sehr vielen Bereichen. Im Folgenden sind nur bestimmte Ausschnitte als Screenshots abgedruckt, um die Übersicht nicht zu verlieren.

Im Bereich *Summary* geben Sie die Bundle ID an, die in iTunes Connect eingerichtet wurde. Bei *Deployment Target* ist die iOS-Version eingestellt, ab der die App laufen soll.

Zum Beispiel ist die Version 5.0 notwendig, um Funktionen des Game Center und der iCloud nutzen zu können.

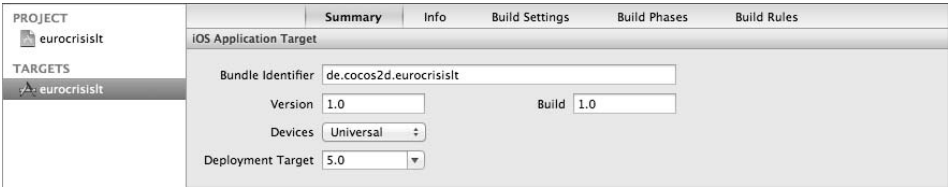


Bild 1.12: Angabe der iOS-Version

Um möglichst viele Geräte im App Store anzusprechen, sollte eine möglichst niedrige iOS-Version angegeben sein. Programmierer sollten immer den Nutzen neu eingeführter Funktionen in aktuellen iOS-Versionen für ihre App gegen den Verbreitungsgrad der Geräte abwägen.

Im Bereich *Info* stehen Variablen mit zugeordneten Werten. Hier sollten *Bundle name*, *Bundle Identifier* und *Bundle display name* für die Anzeige auf dem Display der Geräte richtig angegeben sein.

PROJECT	Summary	Info	Build Settings	Build Phases	Build Rules
eurocrisislt	iOS Application Target				
TARGETS	Custom iOS Target Properties				
eurocrisislt	Key	Type	Value		
	Bundle name	String	Euro Crisis LT		
	Bundle identifier	String	de.cocos2d.eurocrisislt		
	Icon already includes gloss effects	Boolean	YES		
	Bundle display name	String	Euro Crisis LT		
	Status bar is initially hidden	Boolean	YES		

Bild 1.13: Weitere Einstellungen

Bei *Icon already includes gloss effects* sollte *YES* stehen, um zu verhindern, dass das Icon von iOS einen dreidimensionalen Touch durch einen milchigen Überzug bekommt. Das Icon erscheint nun lediglich mit abgerundeten Ecken auf dem Display.

Der Statusbalken mit Angaben wie Uhrzeit und Batterieanzeige stört in Spielen. Daher sollte bei *Status bar is initially hidden* der Wert *YES* stehen.

1.2.4 Icons in eine App integrieren

Eine App für iOS benötigt acht Icons.

Dateiname	Bedeutung / Größe / Gerät
Icon.png	App-Icon in der Größe 57 x 57 Pixel für iPod und iPhone
Icon@2x.png	App-Icon in der Größe 114 x 114 Pixel für iPod und iPhone (Retina-Version)
Icon-72.png	App-Icon in der Größe 72 x 72 Pixel für iPad

Dateiname	Bedeutung / Größe / Gerät
<i>Icon-72@2x.png</i>	App-Icon in der Größe 144 x 144 Pixel für iPad (Retina-Version)
<i>Icon-Small.png</i>	Icon für Suchergebnisse in der Größe 29 x 29 Pixel für iPod und iPhone
<i>Icon-Small@2x.png</i>	Icon für Suchergebnisse in der Größe 58 x 58 Pixel für iPod und iPhone (Retina-Version)
<i>Icon-Small-50.png</i>	Icon für Suchergebnisse in der Größe 50 x 50 Pixel für iPad
<i>Icon-Small-50@2x.png</i>	Icon für Suchergebnisse in der Größe 100 x 100 Pixel für iPad (Retina-Version)

Sobald Sie das App-Icon in den acht Varianten hergestellt haben, kopieren Sie die Dateien in den Ordner mit den Ressourcen.

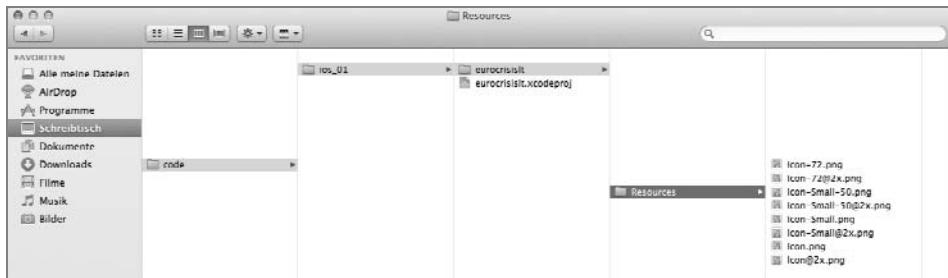


Bild 1.14: Dateien im Ressourcen-Ordner

Auf der linken Seite in Xcode befindet sich ein Ordnerbaum mit Referenzen zu allen Dateien, die zum Projekt gehören. Dort müssen die acht Dateien ebenfalls bei den Resources gelistet sein. Sie können sie per Drag & Drop hinzufügen.

Stichwortverzeichnis

A

Aktion 83, 89
 Android 39, 70
 AVAudioPlayer 109

B

Benutzereingaben 95
 Benutzereinstellungen 110
 Berührungseignis 102
 Beschriftung 116
 Bestenlisten 274
 Bewegung 83
 Bewegungsereignis 98
 Bild 75

C

CCAction 83
 CCActionEase 92
 CCActionInstant 83
 CCActionInterval 83
 CCBezierBy 85
 CCBezierTo 85
 CCBlink 85
 CCCallFuncN 84
 CCDirector 95
 CCFadeIn 85
 CCFadeOut 85
 CCFadeTo 85
 CCFlipX 84
 CCFlipY 84
 CCHide 84
 CCJumpBy 85
 CCJumpTo 85
 CCLabelBMFont 120
 CCLabelTTF 115
 CCLayer 219

CCLayerColor 221
 CCMoveBy 85
 CCMoveTo 85
 CCNode 72
 CCParticleSystem 139
 CCParticleSystemQuad 135
 CCPlace 84
 CCRepeat 90
 CCRepeatForever 90
 CCRotateBy 85
 CCRotateTo 85
 CCScaleBy 85
 CCScaleTo 85
 CCScene 219
 CCSequence 90
 CCShow 84
 CCSkewBy 85
 CCSkewTo 85
 CCSpawn 90
 CCSprite 75
 CCTintBy 85
 CCTintTo 85
 CCToggleVisibility 84
 CCTransitionScene 223
 CCUserDefault 111
 CDSoundEngine 109
 CGPoint 74
 Cocos2D 5, 16
 Cocos2D-API 72
 Cocos2D-X 5, 40
 CocosDenshion 109

D

Drehung 83

E

Eclipse 37
 Effekt 128, 130
 Emitter 130
 Emittertyp 129
 Ereignis 96
 Erfolge 274

G

Game Center 267
 GKAchievement 269
 GKAchievementViewController 272
 GKLeaderboardViewController 271
 GKScore 268
 Glyph Designer 119
 Gravitation 129
 Gravitationseffekte 144

H

Hintergrundmusik 109

I

Icons 25, 44
 iPad 67, 69
 iPhone 67, 69
 iPod 67
 iTunes Connect 18

J

Java 38

K

Koordinatensystem 70, 76

M

Menü 231
 Multitouch 96
 Musik 109

N

NSDictionary 139
 NSMutableArray 250
 NSUserDefaults 110

O

Optionen 241

P

Particle Designer 127

R

Radialeffekte 144
 Retina-Display 67

S

Schlüssel 139
 Schriftart 116
 Schwerkraft 129
 Screen FX 130, 143
 SimpleAudioEngine 109
 Soundeffekt 109
 Spiegelung 83
 Strang 122

T

Teilchensystem 127
 Teilchentextur 128
 Touch 95
 Transformation 77

U

Überblendung 83
 URL 235

W

Website 235
 Winkel 105

X

Xcode 15

Z

Zeichensatz 120
 Zentrum 129

Spiele entwickeln für iOS und Android mit Cocos2D

Mit dem kostenlosen Framework Cocos2D lassen sich Spiele für die großen Smartphone-Plattformen entwickeln. Mehrere Tausend Spiele verwenden bereits die Cocos2D-API. Am Beispiel des Spiels Euro Crisis wird der komplette Ablauf vom Entwurf des Spiels über die Programmierung für iOS und Android bis zur Vermarktung in den App-Stores durchgespielt. Die programmiertechnischen Grundlagen werden ebenso gezeigt wie die einzelnen Schritte der Spielgestaltung.

Profitieren Sie von der Fülle an Klassen, die Cocos2D dem Spieleentwickler zur Verfügung stellt: Szenenmanagement, Sprites, Effekte, Aktionen, Menüs, Teilchensysteme, Textunterstützung, Textur-Atlas und Soundmaschine, punktebasiertes Koordinatensystem zur Nutzung unterschiedlich großer Bildschirme, Eingabe per Finger und Tastatur, Beschleunigungssensor, integrierte Pausen- und Fortsetzungsfunktion und viele mehr.

In diesem Buch finden Sie den Quellcode stets in zwei Varianten: zunächst in Objective-C für die iOS-Entwicklung und immer direkt im Anschluss in C++ für die Android-Variante.



Besuchen Sie unsere Website
www.franzis.de

Aus dem Inhalt:

- Hello Cocos2D
- Xcode und Cocos2D installieren
- Eine App für iOS erstellen
- Eclipse und Cocos2D-X installieren
- Eine App für Android erstellen
- Spiele designen
- Euro Crisis LT als Projekt
- Spielszenen vorbereiten
- Bilder anzeigen
- Aktionstypen, Aktionen starten
- Ereignisse verarbeiten: Touch-Ereignisse, Reaktion auf Bewegungs- und Berührungseignisse
- Sound, Musik abspielen
- Ebenen beschriften
- Teilchensysteme erzeugen: Fliegende Sterne, Regen, Schnee, Explosionen
- Spiellevel gestalten
- Szenen verwalten
- Menüs, Optionen und Schaltflächen
- Bestenlisten

Über die Autoren:

Roland Willms ist Diplom-Mathematiker und veröffentlicht seit über zehn Jahren Fachbücher zu Programmiersprachen, speziell zu Java, C# und Objective-C.

Yann Heeser studiert Informatik und befasst sich professionell mit C++ und der plattformübergreifenden Anwendungsentwicklung für mobile Endgeräte.

Auf www.buch.cd

Der komplette Quellcode der Spiel-App für iOS und Android.



30,- EUR [D]
ISBN 978-3-645-60156-6

FRANZIS