

1 iPhone Packager

Im April 2010, kurz vor der Veröffentlichung der neuen Creative Suite 5 von Adobe, wurden die Lizenzbedingungen von Apple für iPhone-Entwickler dahingehend geändert, dass nur noch von Apple definierte Techniken (C, C++, Objective-C, JavaScript) zur Programmierung von iPhone-Anwendungen verwendet werden dürfen.

Damit verbunden war, dass Adobe die Weiterentwicklung des iPhone Packagers in Flash CS5 einstellte. Auch wenn der iPhone Packager noch mit Flash CS5 ausgeliefert wurde, war die offizielle Empfehlung von Adobe selbst, den Packager für die kommerzielle Anwendungsentwicklung für das iPhone nicht zu verwenden. Zu diesem Zeitpunkt war es bereits mehr als wahrscheinlich, dass Anwendungen, die auf diesem Wege erstellt wurden, nicht für den App Store von Apple zugelassen würden. Aus diesem Grund haben wir uns dagegen entschieden, ein diesbezügliches Kapitel im Flash CS5-Handbuch zu veröffentlichen.

Zeitgleich mit der Drucklegung unseres Buches änderte Apple die Lizenzbedingungen erneut dahingehend, dass die zuvor genannten Einschränkungen wieder aufgehoben wurden. Adobe gab kurz danach bekannt, dass die Weiterentwicklung des iPhone Packagers wieder aufgenommen würde.

Wir konnten zu diesem Zeitpunkt leider in unserem Buch nicht mehr darauf eingehen. So finden Sie im Buch nur wenige Erläuterungen zum iPhone Packager. Daher wird Sie nun dieses Zusatzkapitel über den aktuellen Stand (November 2010) informieren und erläutern, wie Sie den iPhone Packager einsetzen können, um Anwendungen für iOS-Geräte zu entwickeln.

1.1 Voraussetzungen

Bevor Sie planen, eine Anwendung für das iOS mit Flash zu entwickeln, sollten Sie sich zunächst mit den technischen Bedingungen und Einschränkungen sowie den offiziellen Richtlinien für Entwickler von Apple und Adobe beschäftigen.

1.1.1 Einschränkungen

Aufgrund von Änderungen der Lizenzbedingungen seitens Apple wurde die Weiterentwicklung des iPhone Packagers von Adobe für einige Monate eingestellt. Auch aus diesem Grund gibt es bei der Entwicklung von Anwendungen eine Reihe von Einschränkungen, von denen einige in Zukunft wahrscheinlich noch behoben werden. Dazu gehören:

Verbreitung des iOS

Das iOS wurde nach Angaben von Apple vom 1.9.2010 inzwischen auf 120 Millionen Geräten weltweit ausgeliefert und ist damit eines der am weitesten verbreiteten Betriebssysteme für Mobilgeräte.

Adobe AIR SWF-Inhalte für iOS

Adobe AIR SWF-Inhalte lassen sich auch ohne Flash CS5 über den separat erhältlichen Packager for iPhone (<http://labs.adobe.com/technologies/packagerforiphone/>) als iPhone-Anwendung kompilieren. Im vorliegenden Kapitel werden wir ausschließlich auf den in Flash CS5 integrierten iPhone Packager eingehen.

iPhone Packager für andere iOS-Geräte?

Im Moment ist noch nicht abzusehen, ob Adobe die Entwicklung auf alle iOS-Geräte ausweitet. Es ist jedoch durchaus denkbar und mehr als wahrscheinlich.

- **Bildschirm-Auflösung:** Das Format ist noch auf 480x320 Pixel (Querformat) beschränkt. Das iPhone 4 besitzt eine Auflösung von 960x640 Pixel (Querformat). Das iPad von Apple besitzt eine Auflösung von 1.024x768 (Querformat).
- **Dateigröße:** Die Dateigrößen von Anwendungen, die mit dem iPhone Packager erzeugt werden, sind zum jetzigen Zeitpunkt noch deutlich größer als nativ entwickelte Anwendungen.
- **Kompilierungszeit:** Die Kompilierung einer Anwendung mit dem iPhone Packager dauert noch sehr lang, in der Regel mehrere Minuten.
- **Funktionsumfang:** Die übersetzte API von Adobe nutzt nur einen kleinen Teil der offenen API. Das Gyroscope bzw. die Kamera im iPhone 4 kann bisher beispielsweise nicht angesprochen werden.
- **Einreichung (Approval) einer Anwendung für den App Store:** Bisher können Sie eine erzeugte Anwendung nur unter Mac OS X über den sogenannten *Application Loader* zur Einstellung in den App Store einreichen. Der Application Loader ist zurzeit nicht für Windows verfügbar, und es gibt bisher auch keine andere Möglichkeit, eine Anwendung an Apple zu übertragen.
- **Standard-UI-Elemente:** Standardisierte UI-Elemente von iOS-Geräten können über Flash nicht verwendet werden. Möchten Sie Standard-UI-Elemente nutzen, müssen Sie diese selbst implementieren. Hilfreich dafür ist beispielsweise die iPhone GUI PSD-Vorlage von Geoff Teehan, die Sie unter <http://www.teehanlax.com/blog/2010/06/14/iphone-gui-psd-v4/> herunterladen können.

1.1.2 Richtlinien für Entwickler

Die Richtlinien sind auf Englisch für registrierte Apple-Entwickler abrufbar unter <http://developer.apple.com/appstore/guidelines.html>.

Vor der Veröffentlichung der Richtlinien durch Apple wurde bereits auf World of Apple (<http://www.worldofapple.com>) eine inoffizielle Version der Richtlinien (http://images.worldofapple.com/appstoreguidelines_9910.pdf), die der von Apple entspricht, veröffentlicht. Es folgt ein Auszug daraus, nämlich der Abschnitt „Functionality“, sowie eine eigene freie deutsche Übersetzung dieser Richtlinien:

2.1 Apps that crash will be rejected

Anwendungen, die abstürzen, werden abgelehnt.

Eigenverantwortung von Entwicklern

Es unterliegt dem Verantwortungsbereich des Entwicklers, die Richtlinien zu beachten und technische Einschränkungen, die die Entwicklung von iOS-Anwendungen über Flash mit sich bringen, zu beachten. Lesen Sie die Richtlinien deshalb sehr genau.

2.2 Apps that exhibit bugs will be rejected

Anwendungen, die nachweisbare, reproduzierbare Fehler besitzen, werden abgelehnt.

2.3 Apps that do not perform as advertised by the developer will be rejected

Anwendungen, die nicht wie vom Entwickler beworben „performen“, werden abgelehnt.

2.4 Apps that include undocumented or hidden features inconsistent with the description of the app will be rejected

Anwendungen, die versteckte, undokumentierte Eigenschaften, die nicht beschrieben wurden, verwenden, werden abgelehnt.

2.5 Apps that use non-public APIs will be rejected

Anwendungen, die nicht öffentliche APIs verwenden, werden abgelehnt.

2.6 Apps that read or write data outside its designated container area will be rejected

Anwendungen, die Daten außerhalb ihres definierten Containers lesen oder schreiben, werden abgelehnt.

2.7 Apps that download code in any way or form will be rejected

Anwendungen, die Code auf beliebigem Wege oder in beliebiger Form laden, werden abgelehnt.

2.8 Apps that install or launch other executable code will be rejected

Anwendungen, die ausführbaren Code starten oder installieren, werden abgelehnt.

2.9 Apps that are "beta", "demo", "trial", or "test" versions will be rejected

Anwendungen, die sich im „beta“- , „demo“- , „trial“- oder „test“-Stadium befinden, werden abgelehnt.

2.10 iPhone apps must also run on iPad without modification, at iPhone resolution, and at 2X iPhone 3GS resolution

Anwendungen für das iPhone müssen auch auf dem iPad ohne Veränderungen laufen, sowohl mit der iPhone-Auflösung als auch in der 2-fach vergrößerten Auflösung des iPhone 3GS.

2.11 Apps that duplicate apps already in the App Store may be rejected, particularly if there are many of them

Anwendungen, die die Funktionalität von bereits vorhandenen Anwendungen im App Store 1:1 kopieren, werden unter Umständen abgelehnt, insbesondere wenn es schon mehrere dieser Art von Apps gibt.

2.12 Apps that are not very useful or do not provide any lasting entertainment value may be rejected

Anwendungen, die keinem bestimmten sinnvollen Zweck dienen oder keinen unterhaltenden Wert bieten, werden abgelehnt.

2.13 Apps that are primarily marketing materials or advertisements will be rejected

Anwendungen, die primär aus Marketingmitteln oder aus Werbemitteln bestehen, werden abgelehnt.

2.14 Apps that are intended to provide trick or fake functionality that are not clearly marked as such will be rejected

Anwendungen, deren Beabsichtigung es ist, durch Tricks oder Fälschungen Funktionalität zu bieten, ohne diese explizit anzugeben, werden abgelehnt.

2.15 Apps larger than 20 MB in size will not download over cellular networks (this is automatically prohibited by the App Store)

Anwendungen, deren Dateigröße größer als 20 MB ist, können nicht über mobile Datennetze heruntergeladen werden; der Download wird ggf. automatisch vom App Store gesperrt.

2.16 Multitasking apps may only use background services for their intended purposes: VoIP, audio playback, location, task completion, local notifications, etc

Multitaskfähige Anwendungen dürfen nur Hintergrunddienste nutzen, die für einen von Apple definierten Zweck gedacht sind: VoIP, Audio-Wiedergabe, Location (GPS), Anwendungsbeendigung, lokale Benachrichtigungen etc.

2.17 Apps that browse the web must use the iOS WebKit framework and WebKit Javascript

Anwendungen, mit denen man im Web browsen kann, müssen das iOS WebKit Framework und WebKit Javascript verwenden.

2.18 Apps that encourage excessive consumption of alcohol or illegal substances, or encourage minors to consume alcohol or smoke cigarettes, will be rejected

Anwendungen, die dazu anregen, exzessiv Alkohol oder illegale Substanzen zu konsumieren, oder die Minderjährige dazu anregen, Alkohol oder Zigaretten zu konsumieren, werden abgelehnt.

2.19 Apps that provide incorrect diagnostic or other inaccurate device data will be rejected

Anwendungen, die falsche Diagnosen oder andere ungenaue Gerätedaten angeben, werden abgelehnt.

2.20 Developers "spamming" the App Store with many versions of similar apps will be removed from the iOS Developer Program

Entwickler, die den App Store mit sehr vielen Versionen eines Programms „spammen“, werden vom iOS Developer Program ausgeschlossen.

1.2 Vorbereitungen

Bevor Sie mit der Entwicklung einer Anwendung für das iOS beginnen, sind einige Vorbereitungen zu treffen. Dafür ist eine Reihe von einzelnen Schritten erforderlich, die wir Ihnen im Folgenden erläutern werden.

1.2.1 Zertifikat zur Signierung erzeugen

Zunächst müssen Sie sich bei Apple als iOS-Entwickler für das iOS Developer Program registrieren. Um dann selbst Anwendungen erzeugen zu können, benötigen Sie eine kostenpflichtige Mitgliedschaft, die 99 Dollar bzw. 79 Euro pro Jahr kostet. Unter <http://developer.apple.com/programs/ios/> können Sie sich beim iOS Developer Program als Entwickler anmelden.

Nachdem Sie sich registriert haben, benötigen Sie ein Zertifikat, mit dem Sie Ihre Anwendungen signieren. Der Prozess der Zertifikatserstellung ist teilweise nicht ganz einfach. Da es auch hier deutliche Unterschiede bei der Erzeugung zwischen Windows und Mac OS X gibt, werden alle notwendigen Schritte im Folgenden für beide Plattformen getrennt erläutert.

Zertifizierungsprozess unter Mac OS X durchführen

Im folgenden Workshop wird erläutert, wie Sie Ihr Zertifikat zum Signieren von eigenen iOS-Anwendungen unter Mac OS X erzeugen können.

Hinweis: Keine Abbildungen des iOS Dev Center

Leider dürfen wir Ihnen in diesem Abschnitt aus urheberrechtlichen Gründen keine Screenshots aus dem iOS Dev Center zeigen und müssen uns mit den entsprechenden Beschreibungen begnügen.

Empfehlung: Mac OS bevorzugt

Sollten Sie sowohl einen Windows-PC als auch einen Mac besitzen, empfehlen wir Ihnen, den Zertifizierungsprozess unter Mac OS X durchzuführen. Dabei gibt es deutlich weniger Schritte, und der Prozess ist weniger fehleranfällig.

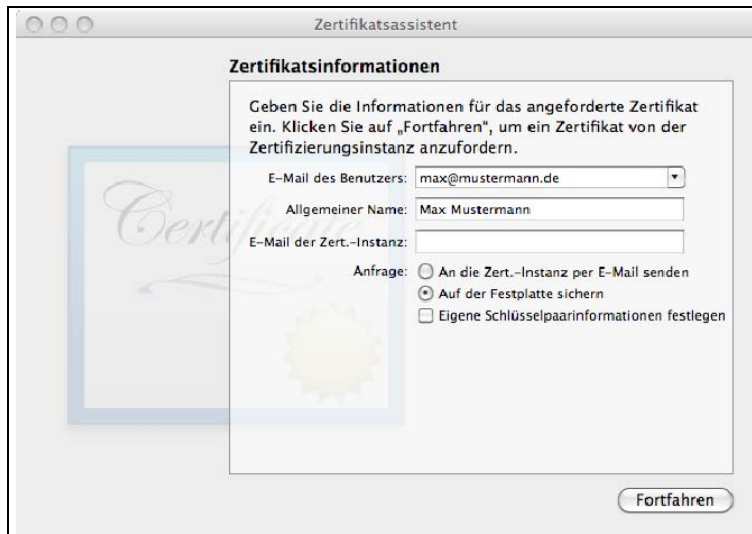


Abbildung 1.1: Zertifikat anfordern

Unter Mac OS X fordern Sie über die Schlüsselbundverwaltung, die Sie unter den Dienstprogrammen finden, über das Menü **SCHLÜSSELBUNDVERWALTUNG • ZERTIFIKATASSISTENT • ZERTIFIKAT EINER ZERTIFIZIERUNGSINSTANZ ANFORDERN** ein neues Zertifikat an. Geben Sie Ihre E-Mail-Adresse und Ihren Namen ein, und aktivieren Sie das Optionfeld **AUF DIE FESTPLATTE SICHERN**.

Klicken Sie auf **FORTFAHREN**, und speichern Sie das Zertifikat auf dem Schreibtisch ab.

Loggen Sie sich mit Ihrem Entwicklerzugang im **iOS Dev Center** ein. Im Bereich **PROVISIONING PORTAL • CERTIFICATES** können Sie die Datei hochladen.

Nach kurzer Wartezeit steht Ihnen dann Ihr Zertifikat im **PROVISIONING PORTAL** im Bereich **CERTIFICATES** zur Verfügung. Klicken Sie neben dem Zertifikat auf **DOWNLOAD**, um das Zertifikat herunterzuladen.

Wählen Sie das signierte Zertifikat aus, und ziehen Sie es per **Drag&Drop** in die Schlüsselbundverwaltung.

1. **Zertifikat in der Schlüsselbundverwaltung erzeugen**
2. **Zertifikat aus dem Apple iOS Dev Center hochladen**
3. **Signiertes Zertifikat herunterladen**
4. **Zertifikat in die Schlüsselbundverwaltung importieren**



Abbildung 1.2: Das Zertifikat wurde importiert.

Selektieren Sie in der Schlüsselbundverwaltung das signierte Zertifikat, und wählen Sie dann aus dem Menü den Menüpunkt **ABLAGE • OBJEKTE EXPORTIEREN** aus. Sie werden daraufhin aufgefordert, ein Kennwort einzugeben. Merken Sie sich dieses gut. Sie benötigen es später noch im iPhone Packager von Flash CS5. Speichern Sie das Zertifikat dann als .p12-Datei ab. Dieses wird dann später in Flash CS5 zur Authentifizierung benötigt. Damit ist der Zertifizierungsprozess unter Mac OS X abgeschlossen.

5. Zertifikat als .p12-Datei exportieren

Zertifizierungsprozess unter Windows durchführen

Im folgenden Workshop wird erläutert, wie Sie Ihr Zertifikat zum Signieren von eigenen iOS-Anwendungen unter Windows erzeugen können. Vorweg sei erwähnt, dass der Zertifizierungsprozess unter Windows etwas umständlicher ist als unter Mac OS X.

Laden Sie unter <http://www.slproweb.com/products/Win32OpenSSL.html> zunächst Visual C++ 2008 Redistributables herunter, und installieren Sie es auf Ihrem System. Laden Sie anschließend auf derselben Seite die für Ihr System geeignete Version von OpenSSL herunter, und installieren Sie sie. Sollten Sie eine 64-Bit-Version von Windows benutzen, empfehlen wir Ihnen, dennoch die 32-Bit-Version zu installieren. Unter Windows 7 (64 Bit) kam es bei der Erzeugung des Zertifikats mit der 64-Bit-Version von OpenSSL zu Fehlern kommen.

1. Visual C++ 2008 Redistributables und OpenSSL herunterladen und installieren

Öffnen Sie die Windows-Eingabeaufforderung unter Windows, z.B. indem Sie den Menüpunkt **START • ALLE PROGRAMME • ZUBEHÖR • EINGABEAUFFORDERUNG** öffnen. Wechseln Sie in das bin-Verzeichnis von OpenSSL, z.B. über `cd \openssl\bin`. Geben Sie folgenden Befehl ein, um einen privaten Schlüssel zu erzeugen:

2. Privaten Schlüssel mit OpenSSL erzeugen

```
openssl genrsa -out mykey.key 2048
```

Es kann an diesem Punkt vorkommen, dass eine Warnung oder ein Fehlerhinweis erscheint. Zunächst sollten Sie dann noch einmal die Syntax des Befehls überprüfen. Abweichend von Adobes Dokumentation können Sie versuchen, trotz Fehler- oder Warnhinweisen an dieser Stelle, den Prozess fortzusetzen. Sie sollten allerdings unbedingt überprüfen, ob der Schlüssel erzeugt wurde und die Datei größer als 0 Bytes ist. Ob der Schlüssel erzeugt wurde, können Sie leicht überprüfen. Geben Sie `dir` ein, und schauen Sie, ob eine `mykey.key`-Datei erzeugt wurde.

```

Administrator: C:\Windows\system32\cmd.exe
22.09.2010 12:38 <DIR> .
22.09.2010 12:38 <DIR> ..
22.09.2010 12:38 <DIR> bin
01.06.2010 06:31 407.933 changes.txt
22.09.2010 12:38 <DIR> exp
01.06.2010 06:31 43.836 faq.txt
22.09.2010 12:38 <DIR> include
22.09.2010 12:38 <DIR> lib
05.06.2010 17:49 1.565.184 libeay32.dll
05.06.2010 17:49 307.200 libssl32.dll
27.05.2008 11:41 6.279 license.txt
01.06.2010 06:17 23.910 news.txt
19.05.2004 16:22 30.423 OpenSSLhelp.chm
01.06.2010 06:31 9.129 readme.txt
05.06.2010 17:49 307.200 sslseay32.dll
22.09.2010 12:38 19.095 unins000.dat
22.09.2010 12:37 721.694 unins000.exe
11 Datei(en), 3.441.883 Bytes
6 Verzeichnis(se), 10.667.614.208 Bytes frei

C:\OpenSSL64>cd bin
C:\OpenSSL64\bin>openssl genrsa -out mykey.key 2048
WARNING: can't open config file: C:\OpenSSL\bin\openssl.cfg
Loading 'screen' into random state - done
Generating RSA private key, 2048 bit long modulus
.....+++
-----+-----
e is 65537 <0x10001>
C:\OpenSSL64\bin>_

```

Abbildung 1.3: Der private Schlüssel wurde erzeugt.

Geben Sie in der Eingabeaufforderung den folgenden Befehl ein. Fügen Sie dabei Ihre E-Mail-Adresse, Ihren Namen und gegebenenfalls den Ländercode Ihres Landes (DE) an den entsprechenden Stellen ein:

```

openssl req -new -key mykey.key -out
CertificateSigningRequest.certSigningRequest -subj
"/emailAddress=yourAddress@example.com, CN=John
Doe, C=US"

```

Überprüfen Sie, ob im `bin`-Verzeichnis von OpenSSL eine Datei mit dem Namen „CERTSIGNINGREQUEST-Datei (.certSigningRequest)“ erzeugt wurde.

Loggen Sie sich mit Ihrem Entwicklerzugang im iOS Dev Center ein. Im Bereich PROVISIONING PORTAL • CERTIFICATES können Sie die Datei hochladen. Klicken Sie dazu auf REQUEST CERTIFICATE.

Nach kurzer Wartezeit steht Ihnen dann Ihr Zertifikat im PROVISIONING PORTAL im Bereich CERTIFICATES zur Verfügung. Klicken Sie neben dem Zertifikat auf DOWNLOAD, um das Zertifikat herunterzuladen. Sie erhalten eine .cer-Datei

3. Zertifikat erzeugen

4. Zertifikat aus dem Apple iOS Dev Center hochladen

5. Signiertes Zertifikat herunterladen

„developer_identity.cer“, die im nächsten Schritt noch in ein .p12-Datei umgewandelt werden muss.

Speichern Sie die .cer-Datei zunächst in das bin-Verzeichnis von OpenSSL. Führen Sie dann im bin-Verzeichnis folgenden Befehl aus:

```
openssl x509 -in developer_identity.cer -inform DER  
-out developer_identity.pem -outform PEM
```

Sollten Sie keine Fehlermeldung erhalten, ist alles in Ordnung. Führen Sie dann folgenden Befehl aus:

```
openssl pkcs12 -export -inkey mykey.key -in  
developer_identity.pem -out iphone_dev.p12
```

Geben Sie Ihr Passwort, das Sie bei der Erzeugung des Zertifikats angegeben haben, ein.

Sollten Sie keine Fehlermeldung erhalten, ist der Prozess damit abgeschlossen. Im bin-Verzeichnis von OpenSSL sollte sich die Datei „iphone_dev.p12“ befinden, die Sie später in Flash CS5 zur Signierung Ihrer Anwendungen nutzen können. Erfahrungsgemäß funktioniert der vorher genannte Prozess nicht beim ersten Mal. Sollten Sie Probleme bei der Erstellung des Zertifikats haben, können Sie eventuell versuchen, das Zertifikat über die Software „BirdKey“ von SBA zu erstellen. Wir haben diese Software nicht verwendet, allerdings scheinen damit einige User mehr Erfolg zu haben.

6. Umwandlung der developer_identity.cer in iphone_dev.p12

7. Fertig!

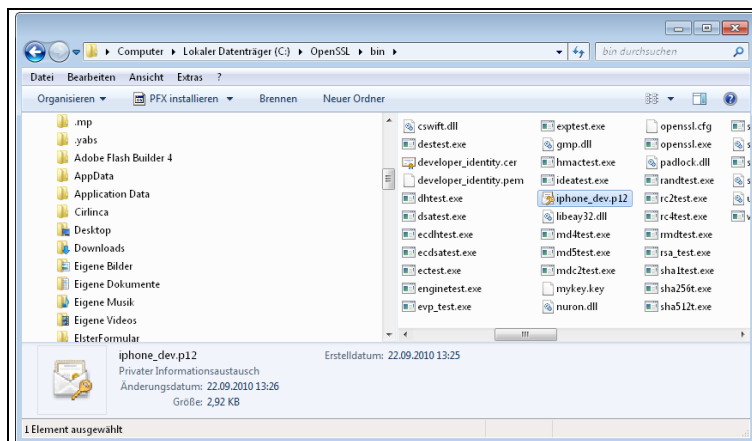


Abbildung 1.4: Das Zertifikat wurde erzeugt.

1.2.2 Gerät einrichten

Um eine Anwendung auf einem Testgerät installieren zu können, müssen Sie das jeweilige Gerät zunächst im iOS Dev Center registrieren. Loggen Sie sich dazu in das iOS Dev Center ein, wechseln Sie in den Bereich IOS PROVISIONING PORTAL, und klicken Sie im linken Menü auf DEVICES. Klicken Sie dann auf ADD DEVICES, um ein neues Gerät zu registrieren. Geben Sie im Feld DEVICE NAME den Bezeichner Ihres

Geräts ein. Sie finden den Bezeichner Ihres iPhones, wenn Sie es mit iTunes verbinden. Wählen Sie Ihr Gerät in iTunes auf der linken Seite aus. Im Bereich ÜBERSICHT wird Ihnen der Name angezeigt. Klicken Sie bei gedrückter (STRG)-Taste auf das Feld SERIENNUMMER, um sich die Device ID (40-Hex-Ziffer) anzeigen zu lassen (siehe Abbildung 1.5).



Abbildung 1.5: Geräteinformationen in iTunes 10

Sie können dann unter Windows via (STRG)+(C) die Device ID in die Zwischenablage kopieren und im iOS Dev Center in das entsprechende Feld mit (STRG)+(V) einfügen. Klicken Sie auf SUBMIT, nachdem Sie beide Felder ausgefüllt haben, um das Gerät zu registrieren.

1.2.3 App-ID

Für jede Anwendung, die Sie erstellen möchten, benötigen Sie zunächst eine App-ID. Loggen Sie sich dazu zunächst in das iOS Dev Center ein, und wechseln Sie dann in den Bereich IOS PROVISIONING PORTAL. Klicken Sie dort im linken Menü auf APP-ID, um eine neue App-ID anzulegen. Füllen Sie die Textfelder in dem Formular aus, und klicken Sie anschließend auf SUBMIT.

1.2.4 Provisioning-Profil

Damit Sie Ihre Anwendung kompilieren und auf Ihrem Gerät testen können, benötigen Sie ein sogenanntes *Provisioning-Profil*. Das Profil wird beim Kompilieren der Anwendung von Flash CS5 benötigt. Sie müssen es zusätzlich auf Ihrem iPhone installieren, um eigene Anwendungen auf das Gerät installieren zu können.

Loggen Sie sich dazu zunächst in das iOS Dev Center ein, und wechseln Sie in den Bereich IOS PROVISIONING PORTAL. Klicken Sie dann links auf PROVISIONING und anschließend auf die Schaltfläche NEW PROFILE.

Füllen Sie die Formularfelder aus. Unter APP-ID wählen Sie die zuvor erstellte App-ID aus. Klicken Sie dann auf SUBMIT, um das Profil zu erstellen. Nach kurzer Wartezeit sollte Ihr Provisioning-Profil im Bereich PROVISIONING zum Download bereitstehen. Laden Sie es daraufhin herunter.

Um das Provisioning-Profil auf Ihrem Gerät zu installieren, gehen Sie wie folgt vor: Öffnen Sie iTunes, und verbinden Sie Ihr Gerät damit. Klicken Sie im Menü auf DATEI • DATEI ZUR MEDIATHEK HINZUFÜGEN, und wählen Sie Ihr Provisioning-Profil aus. Um zu überprüfen, ob das Provisioning-Profil in Ihre Mediathek aufgenommen wurde, können Sie den Vorgang wiederholen. Es sollte dann ein Hinweis erscheinen, dass sich das Profil bereits in der Mediathek befindet, und Sie werden gefragt, ob es ersetzt werden soll. Nachdem das Profil zu Ihrer Mediathek hinzugefügt wurde, müssen Sie das iPhone mit iTunes synchronisieren, damit das Profil auf das iPhone übertragen werden kann. Damit sind die Vorbereitungen abgeschlossen.

Distribution Provisioning Profile

Sollten Sie eine Anwendung bei Apple einreichen, um diese im App Store verfügbar zu machen, benötigen Sie dafür ein sogenanntes *Distribution Provisioning Profile*, das Sie auf dieselbe Art und Weise im iOS Dev Center unter PROVISIONING unter dem Reiter DISTRIBUTION erstellen können.

1.3 Kompilierung

Für diesen Abschnitt haben wir eine einfache Anwendung geschrieben, die Sie auch für Testzwecke nutzen können.

Beispiel

iPhone_Packager\Tiny_Memory\
Tiny Memory.fl



Abbildung 1.6: Das beiliegende „Tiny Memory“ auf einem iPhone 3GS

Es handelt sich um ein Memory-Spiel namens *Tiny Memory*, das bereits von Apple zugelassen und im App Store kostenlos zur Verfügung steht. Sie können es selbst unter <http://itunes.apple.com/de/app/tiny-memory/id398365289?mt=8> herunterladen. Die Quelldateien finden Sie auf der Bonus-Seite dieses Buchs zum Download.

Damit Sie wissen, welche Einstellungen Sie beim Kompilieren vornehmen können, öffnen Sie zunächst die beiliegende Quelldatei *Tiny Memory.fl*. Öffnen Sie dann über das Menü DATEI • IPHONE OS EINSTELLUNGEN das Fenster IPHONE BETRIEBSSYSTEMEINSTELLUNGEN.

1.3.1 Allgemein

Unter dem Reiter ALLGEMEIN (siehe Abbildung 1.7) finden Sie folgende Einstellungen:

Hinweis: Sounddateien

Die in der Originalversion des Spiels verwendeten Sounddateien können wir Ihnen in dieser Beispielversion aufgrund von Lizenzbedingungen leider nicht zur Verfügung stellen. Alle Soundfunktionen wurden deshalb auch im Quellcode auskommentiert.

- **AUSGABEDATEI:** Die Ausgabedatei mit der Dateierweiterung IPA ist die Datei, die Sie zum Testen auf Ihr iOS-Gerät und später auch an Apple übertragen.
- **APP-NAME:** Der Name der Anwendung, wie sie daraufhin auch auf Ihrem iOS-Gerät erscheint.
- **VERSION:** Die Versionsnummer der Anwendung. Wenn Sie später eine Anwendung in den App Store stellen, sollte diese Versionsnummer mit der Versionsnummer im App Store übereinstimmen.
- **SEITENVERHÄLTNIS:** Hier können Sie festlegen, ob Ihre Anwendung standardmäßig im Hoch- oder im Querformat gestartet werden soll.

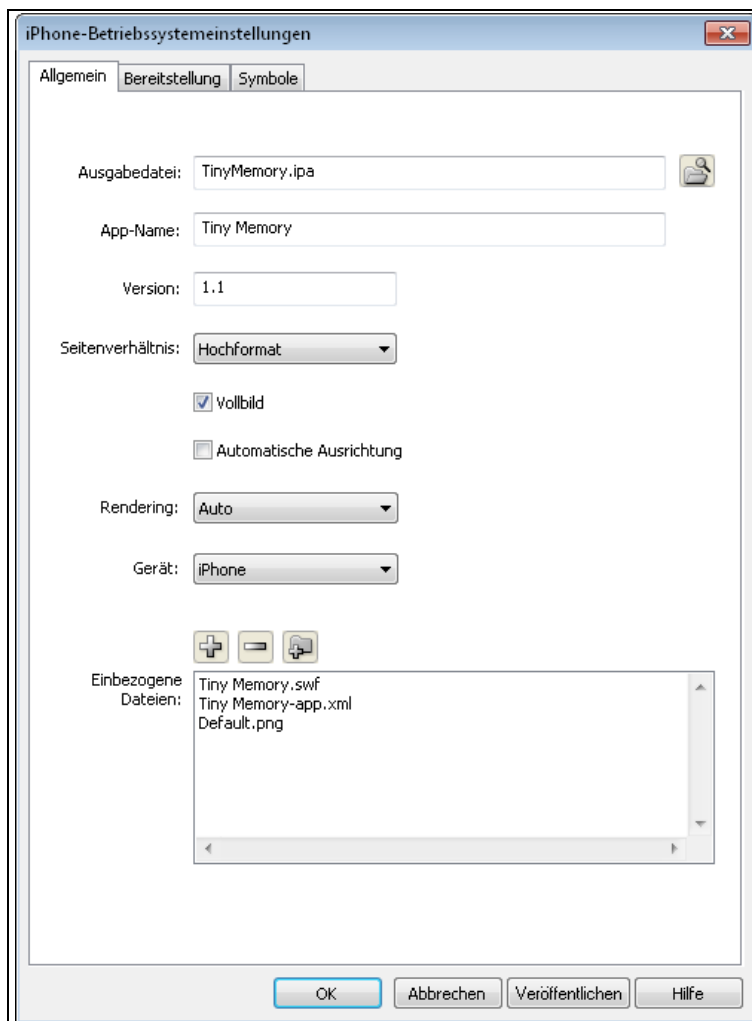


Abbildung 1.7: Der Reiter ALLGEMEIN im Überblick

- **VOLLBILD:** Im Normalfall sollten Sie diese Option aktivieren. Ist sie deaktiviert, nutzt die Anwendung nicht die volle Bildschirmgröße, und grafische Elemente werden skaliert (siehe Abbildung 1.8).

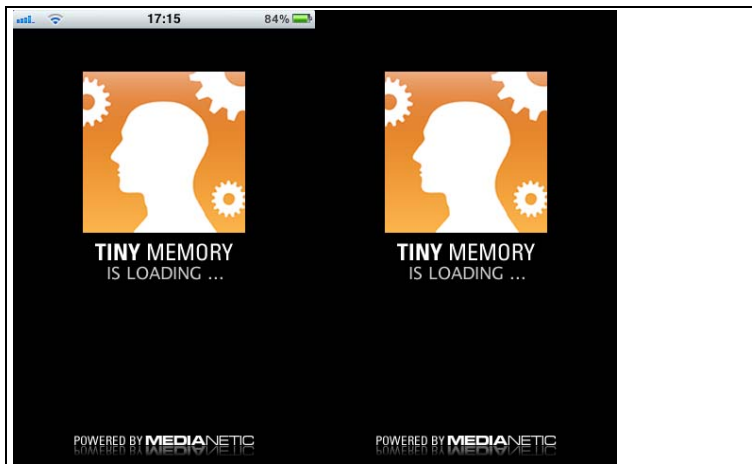


Abbildung 1.8: Links deaktiviertes, rechts aktiviertes Vollbild

- **AUTOMATISCHE AUSRICHTUNG:** In der Regel sollten Sie diese Option deaktivieren. Ist die Option aktiviert, wird Ihre Anwendung je nach Ausrichtung des Geräts im Hoch- oder Querformat dargestellt. Beachten Sie dabei, dass Sie die Positionierung und Skalierung aller UI-Elemente dazu selbst innerhalb der Anwendung vornehmen müssen (siehe dazu auch Abbildung 1.9).

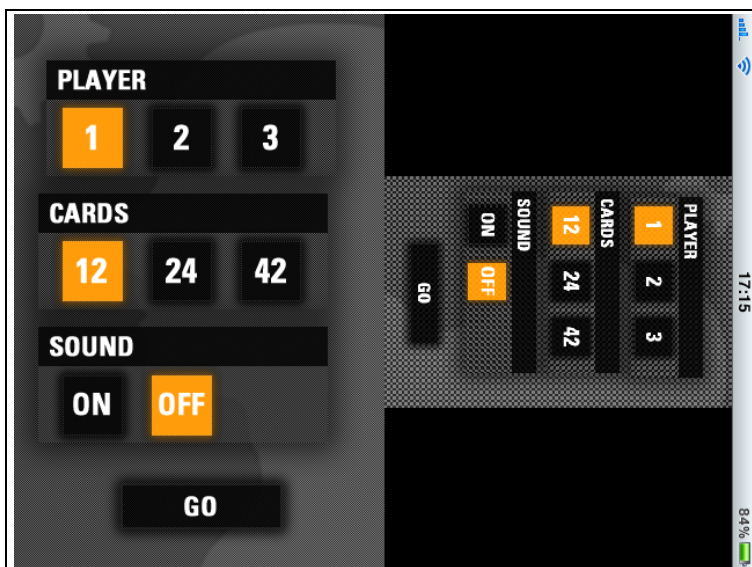


Abbildung 1.9: Links: Hochformat, rechts: die unveränderte Version im Querformat

- **RENDERING:** Belassen Sie die Einstellung auf AUTO. Die Einstellung CPU sorgt dafür, dass Berechnungen an die CPU des Geräts übergeben werden. Sollten Sie eine besonders grafiklastige Anwendung entwickeln, können Sie mit der Einstellung GPU die Berechnungen an den Grafikprozessor des Geräts delegieren.

- **GERÄT:** Wählen Sie hier das Gerät aus, für das Sie die Anwendung entwickeln möchten.
- **EINBEZOGENE DATEIEN:** In diesem Bereich können Sie Dateien hinzufügen, die mit der Anwendung ausgeliefert werden sollen. Standardmäßig finden Sie hier eine SWF-Datei und eine XML-Datei, die Metainformationen wie die Versionsnummer enthält. Manuell hinzufügen sollten Sie eine 320x480 Pixel (iPhone) große Datei *default.png*. Diese wird beim Starten der Anwendung angezeigt. Sie sollten diese Datei im Root-Verzeichnis Ihrer Anwendung und nicht in einem Unterverzeichnis anlegen.

1.3.2 Bereitstellung

Unter dem Reiter **BEREITSTELLUNG** stehen Ihnen die in Abbildung 1.10 dargestellten Einstellungen zur Verfügung.

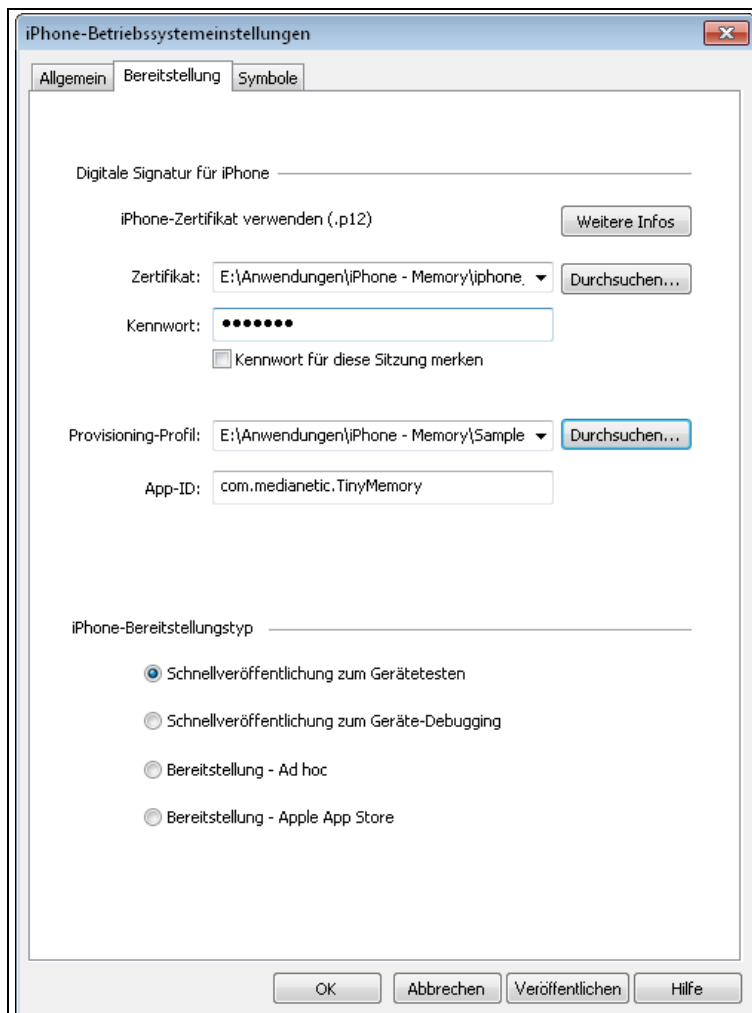


Abbildung 1.10: Der Reiter **BEREITSTELLUNG** im Überblick

- **ZERTIFIKAT:** Wählen Sie hier das p12-Zertifikat aus, das Sie zuvor erstellt hatten. Achten Sie darauf, dass Sie für

den jeweiligen Zweck das richtige Zertifikat auswählen. Zum Testen der Anwendung auf einem eigenen Gerät benötigen Sie ein anderes Zertifikat als das zur Bereitstellung in den App Store.

- **KENNWORT:** Geben Sie hier das Kennwort ein, das Sie bei der Zertifikatserstellung verwendet haben.
- **PROVISIONING-PROFIL:** Wählen Sie das erstellte Provisioning-Profil aus, und achten Sie dabei, wie auch beim Zertifikat selbst, auf die richtige Version.
- **APP-ID:** Tragen Sie hier die App-ID ein, die Sie zuvor im iOS Dev Center angelegt haben.
- **IPHONE-BEREITSTELLUNGSTYP – SCHNELLVERÖFFENTLICHUNG ZUM GERÄTETESTEN:** Verwenden Sie diese Einstellung, wenn Sie die Anwendung zum Testen auf Ihrem iOS-Gerät verwenden möchten.
- **IPHONE-BEREITSTELLUNGSTYP – SCHNELLVERÖFFENTLICHUNG ZUM GERÄTE-DEBUGGING:** Wählen Sie diese Einstellung, wenn Sie die Anwendung auf Ihrem iOS-Gerät zum Testen installieren möchten und zudem den Flash Debugger nutzen möchten, um beispielsweise trace-Ausgaben zu kontrollieren.
- **IPHONE-BEREITSTELLUNGSTYP – BEREITSTELLUNG – AD HOC:** Möchten Sie Ihre Anwendung anderen für Beta-Testzwecke bereitstellen, können Sie diese Option wählen, um bis zu 100 Benutzern Ihre Anwendung zu Testzwecken bereitzustellen. Sie müssen die Benutzer zusätzlich im iOS Dev Center registrieren.
- **IPHONE-BEREITSTELLUNGSTYP – BEREITSTELLUNG – APPLE APP STORE:** Nachdem Sie Ihre Anwendung getestet haben, wählen Sie diese Option, um die Anwendung für die Bereitstellung in den App Store zu kompilieren.

1.3.3 Symbole

Unter dem Reiter SYMBOLE definieren Sie die Programmsymbole, die Sie für Ihre Anwendung im iOS-Gerät nutzen möchten. Sie benötigen mindestens jeweils ein Symbol in den Größen:

- **29x29 Pixel:** Die Spotlight-Suche verwendet dieses Icon.
- **57x57 Pixel:** Das ist das Programm-Icon, das im Home-Bildschirm des iOS-Geräts verwendet wird.
- **512x512 Pixel:** Die Grafik wird für die Darstellung in iTunes für Ihre Anwendung verwendet. Die angegebene Version wird nur für Testversionen Ihrer Anwendung verwendet. Stellen Sie Ihre fertige Anwendung in den App Store ein, sollten Sie die Grafik im JPEG-Format laut Do-

kumentation separat mitliefern. Wir haben diese Version nicht erstellt, offensichtlich ist diese optional.

- 48x48 Pixel: Diese Version wird in Spotlight im iPad verwendet.
- 72x72 Pixel: Diese Version wird für den Home-Bildschirm im iPad genutzt.

1.4 Übertragung

Nachdem Sie eine Anwendung unter Angabe des entsprechenden Provisioning-Profiles und des Zertifikats kompiliert haben, können Sie die Anwendung über iTunes auf Ihr iOS-Gerät übertragen. Wie zuvor beschrieben, müssen Sie dazu zunächst das Provisioning-Profil in die Mediathek von iTunes einfügen. Sie sollten dann zunächst das iOS-Gerät synchronisieren. Im iPhone können Sie unter EINSTELLUNGEN • ALLGEMEIN ganz unten sehen, ob das Profil auf dem Gerät installiert wurde.



Abbildung 1.11: Das Provisioning-Profil „Sampleprofile“ ist auf dem iPhone installiert.

Als Nächstes fügen Sie Ihre erzeugte Anwendung in iTunes ein. Sie können die IPA-Datei dazu einfach per Drag&Drop in iTunes kopieren. Die Anwendung sollte dann in der Mediathek unter APPS angezeigt werden.

Alte Version vorher löschen
Möchten Sie eine neue Version der Anwendung aufspielen, sollten Sie die alte Anwendung vorher in iTunes unter MEDIATHEK • APPS entfernen.



Abbildung 1.12: „Tiny Memory“ wurde nach iTunes übertragen.

Überprüfen Sie dann Ihre Synchronisationseinstellungen, und stellen Sie sicher, dass die Anwendung zur Synchronisation ausgewählt wurde. Klicken Sie dazu in iTunes auf Ihr iOS-Gerät, und wählen Sie dann den Reiter APPS aus.



Abbildung 1.13: „Tiny Memory“ wurde zur Synchronisation mit dem Gerät ausgewählt.

Klicken Sie dann auf SYNCHRONISIEREN. Wurde Ihre Anwendung korrekt kompiliert und haben Sie das richtige Provisioning-Profil ausgewählt, sollte die Anwendung daraufhin auf Ihr Gerät übertragen werden.

1.5 ActionScript API

Sie haben jetzt gelernt, wie Sie eine Anwendung mithilfe des iPhone Packagers einrichten und auf Ihrem Gerät installieren können. Auf die Programmierung einer Anwendung wurde bisher nur beiläufig eingegangen. Als ersten Einstieg können Sie die beiliegende Anwendung nutzen, sie nutzt allerdings nur wenige gerätespezifische Funktionen.

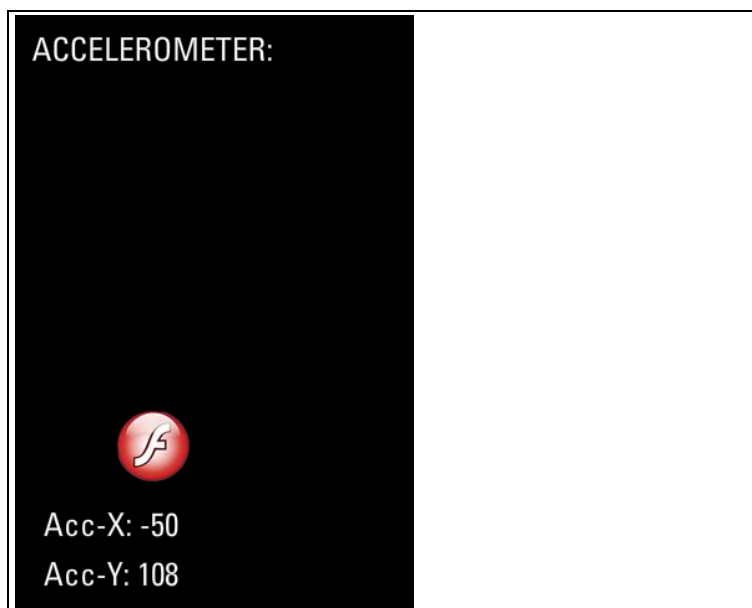
Grundsätzlich können Sie die ActionScript 3 API verwenden, die Sie auch zur Erstellung von Anwendungen für den Flash Player oder AIR-Anwendungen nutzen. Dabei gibt es einige Einschränkungen, lesen Sie dazu auch das Kapitel 3 der offiziellen Dokumentation von Adobe (http://download.macromedia.com/pub/labs/packagerforiphone/packagerforiphone_devguide.pdf).

Davon abgesehen, gibt es eine Reihe von Funktionen der ActionScript 3 API, die speziell für mobile Endgeräte gedacht sind. Es folgt jeweils eine kurze Erläuterung sowie ein Code-Beispiel.

1.5.1 Accelerometer API

Mit der Accelerometer-Klasse können Sie Werte des Beschleunigungssensors abfragen. So könnten Sie beispielsweise ein Spiel entwickeln, das über den Beschleunigungssensor gesteuert wird.

Beispiel



App-ID und Provisioning-Profil

Um die folgenden Demos auf Ihrem Gerät installieren zu können, müssen Sie jeweils eine App-ID und ein Provisioning-Profil im iOS Dev Center erstellen.

Beispiel

iPhone_Packager\Accelerometer\
AccelerometerSample.flas

Abbildung 1.14: Die Demo auf einem iPhone 3GS

In Flash CS5 wird zunächst über DATEI • NEU • IPHONE OS ein neues Dokument erstellt. In diesem Beispiel wird als Erstes eine Grafik *flashSign.png* importiert und in einen MovieClip umgewandelt. Dem MovieClip wird der Instanzname `flashSign_mc` zugewiesen. Der MovieClip wird anschließend auf der Bühne platziert. Weiterhin werden zwei dynamische Textfelder mit den Instanznamen `xPos_txt` und `yPos_txt` erstellt und im unteren Bereich der Bühne platziert. Auf der ersten Ebene wird dem ersten Schlüsselbild der Code der Beispielanwendung zugewiesen:

```

import flash.sensors.Accelerometer;
import flash.events.AccelerometerEvent;
import flash.events.Event;
if (Accelerometer.isSupported) {
    var acc:Accelerometer = new Accelerometer();
    acc.addEventListener(AccelerometerEvent.
        UPDATE,updatePosition);
    var tx:int = 0;
    var ty:int = 0;
    this.addEventListener(Event.ENTER_FRAME,
        resetPos);
}
function updatePosition(e:AccelerometerEvent):void
{
    tx = e.accelerationX * 200;
    ty = e.accelerationY * 200;
}
function resetPos(e:Event):void {
    flashSign_mc.x = (stage.stageWidth/2)+tx;
    flashSign_mc.y = (stage.stageHeight/2)+ty;
    xPos_txt.text = "Acc-X: " + String(tx);
    yPos_txt.text = "Acc-Y: " + String(ty);
}

```

Der Code sorgt dafür, dass die Grafik ausgehend vom Mittelpunkt der Bühne, je nach Position des Geräts (Beschleunigungssensor), nach links oder rechts (`accelerationX`) und nach oben oder unten (`accelerationY`) verschoben wird.

1.5.2 Geolocation API

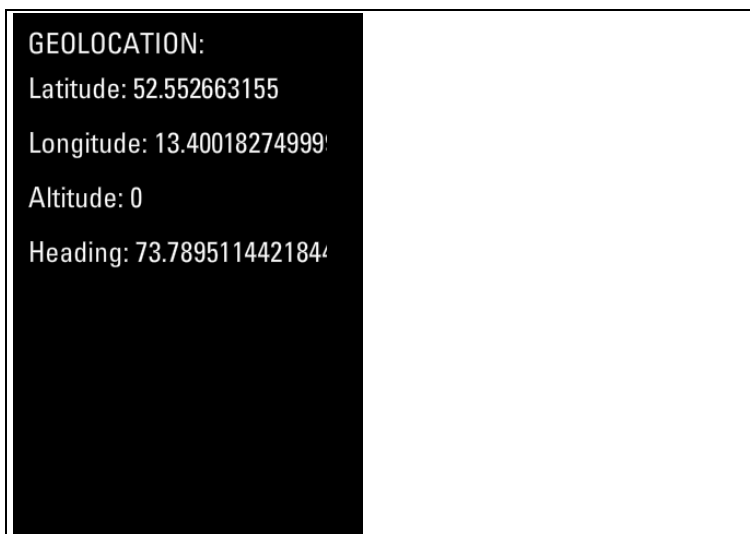


Abbildung 1.15: Die Demo auf einem iPhone 3GS

Über die Geolocation-Klasse können Sie Werte des GPS-Empfängers des Geräts abfragen, z.B. um die aktuelle Position (geografische Länge und Breite) des Geräts zu bestimmen und in einer Kartenanwendung sichtbar zu machen.

Beispiel

```
import flash.sensors.Geolocation;
import flash.events.GeolocationEvent;
if (Geolocation.isSupported) {
    var myGeo:Geolocation = new Geolocation();
    myGeo.addEventListener(GeolocationEvent.
        UPDATE, onGeoUpdate);
    myGeo.setRequestedUpdateInterval(50);
}
function onGeoUpdate(e:GeolocationEvent):void {
    lat_txt.text = "Latitude:
"+String(e.latitude);
    long_txt.text = "Longitude:
"+String(e.longitude);
    alt_txt.text = "Altitude:
"+String(e.altitude);
    heading_txt.text = "Heading:
"+String(e.heading);
}
```

Beispiel

```
iPhone_Packager\Geolocation\
GeolocationSample fla
```

1.5.3 Touch, Multitouch und Gesture API

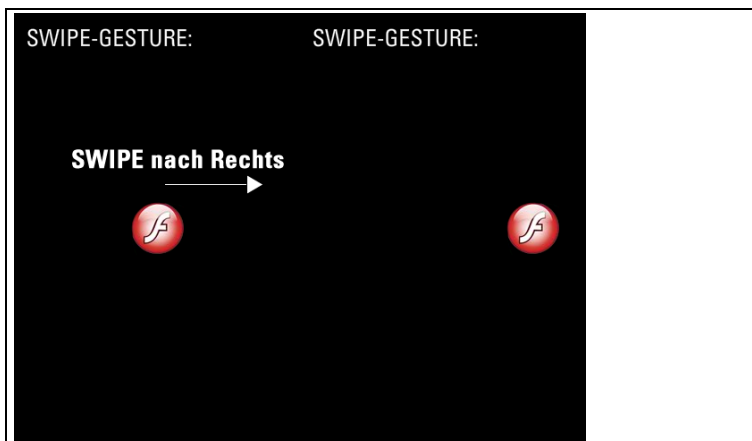


Abbildung 1.16: Die Demo auf einem iPhone 3GS. Hier wurde eine Swipe-Geste nach rechts ausgeführt.

Mithilfe der Multitouch-Klasse können Sie sowohl auf Multitouch-Ereignisse als auch auf spezifische Gesten (wie z.B. die Swipe-Geste) reagieren, um die Interaktion Ihrer Anwendung zu definieren.

Beispiel

```
import flash.ui.Multitouch;
import flash.events.GestureEvent;
Multitouch.inputMode = MultitouchInputMode.GESTURE;
flashSign_mc.addEventListener(TransformGestureEvent
    .GESTURE_SWIPE , swipeMe);
function swipeMe(e:TransformGestureEvent):void {
    if (e.offsetX == 1) {
        // Nach Rechts
        flashSign_mc.x += 100;
    }
    if (e.offsetX == -1) {
        // Nach Links
        flashSign_mc.x -= 100;
    }
    if (e.offsetY == 1) {
        // Nach unten
        flashSign_mc.y += 100;
    }
    if (e.offsetY == -1) {
        // Nach oben
        flashSign_mc.y -= 100;
    }
}
```

Der MovieClip `flashSign_mc` wird je nach Swipe-Richtung in die jeweilige Richtung um 100 Pixel verschoben.

Beispiel

`iPhone_Packager\SwipeGesture\GestureSample fla`

1.6 Ressourcen

Abschließend sind im Folgenden einige der wichtigsten Quellen aufgelistet, die Ihnen wertvolle Informationen bieten und Ihnen dabei weiterhelfen, sich noch intensiver mit dem Thema auseinanderzusetzen.

1.6.1 Offizielle Dokumentationen

- **iPhone Packager Developer Guide von Adobe:**
http://download.macromedia.com/pub/labs/packagerfor-iphone/packagerforiphone_devguide.pdf
- **iPhone Packager Developer FAQ von Adobe:**
http://labs.adobe.com/wiki/index.php/Packager_for_iPhone:Developer_FAQ
- **iPhone Packager Release Notes von Adobe:**
http://labs.adobe.com/wiki/index.php/Packager_for_iPhone:Release_Notes

- **iOS Application Programming Guide von Apple:**

<http://developer.apple.com/library/ios/#documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/Introduction/Introduction.html>

1.6.2 iPhone Packager Forum

Suchen Sie Hilfe oder haben Sie eine Frage zu einer konkreten Problemstellung, dann empfehlen wir Ihnen das iPhone Packager Forum von Adobe, das Sie über *<http://forums.adobe.com/community/labs/packagerforiphone/>* erreichen.