Aktualisierte Version zum iPhone-SDK 2.1

Im folgenden wird erklärt wie Ihr mit Hilfe von Xcode und dem iPhone-Simulator eine erstes Programm names "HelloWorld" erstellen könnt. Ihr benötigt dazu eine Entwicklungsumgebung. Ein Anleitung dazu habe ich hier geschrieben.

Wir werden also ein kleines Programm schreiben, welches ein weißes Fenster öffnet und in diesem eine Navigationsleiste darstellt. Diese Navigationsleiste werden wir dann mit dem Titel "Hello World" versehen. Sollten Ihr Probleme haben, findet Ihr das "HelloWorld" Projekt am Ende des Tutorials als Download zum selber probieren.

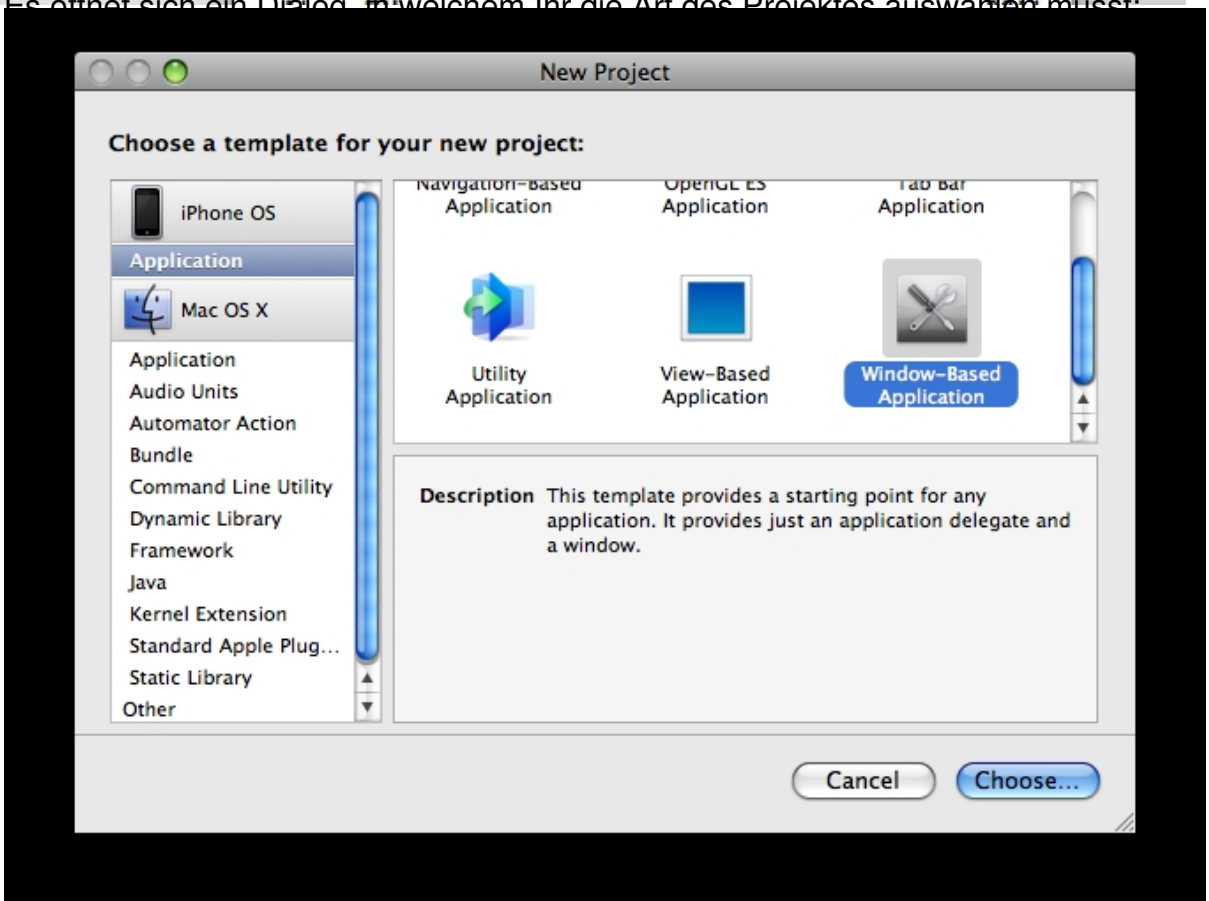
1. Projekt erstellen

Öffnet bitte XCode 3 (standard mässig zu finden unter /Developer/Applications/Xcode.app). Ihr werdet wahrscheinlich mit folgendem Dialog begrüßt:

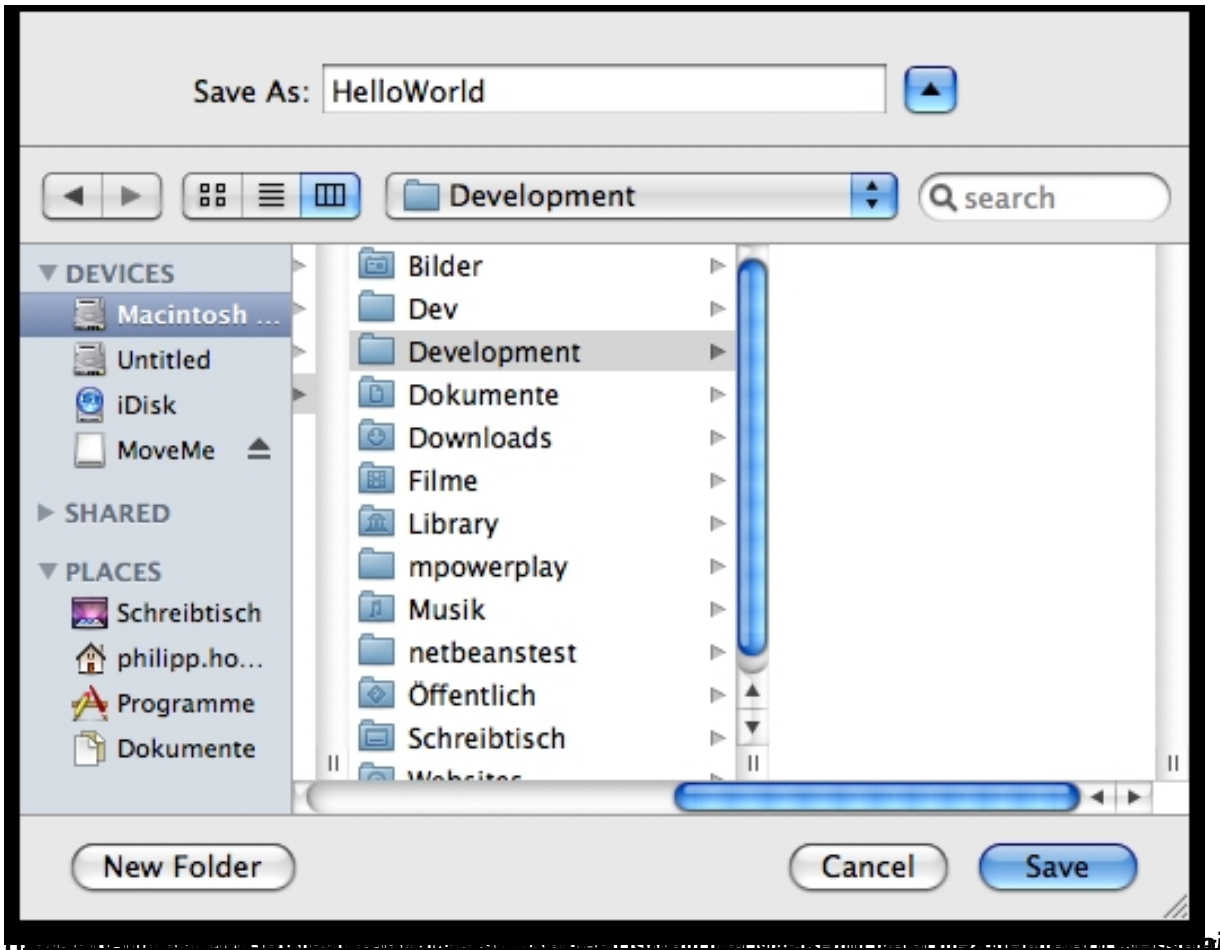
Wenn Ihr nicht mehr wollt, das dieses Begrüßungsfenster bei jedem Start von Xcode erscheint, entfernt Ihr bitte einfach das Häckchen bei "Show at launch" im linken untern Bereich des Fensters. Schliesst jetzt das Fenster.

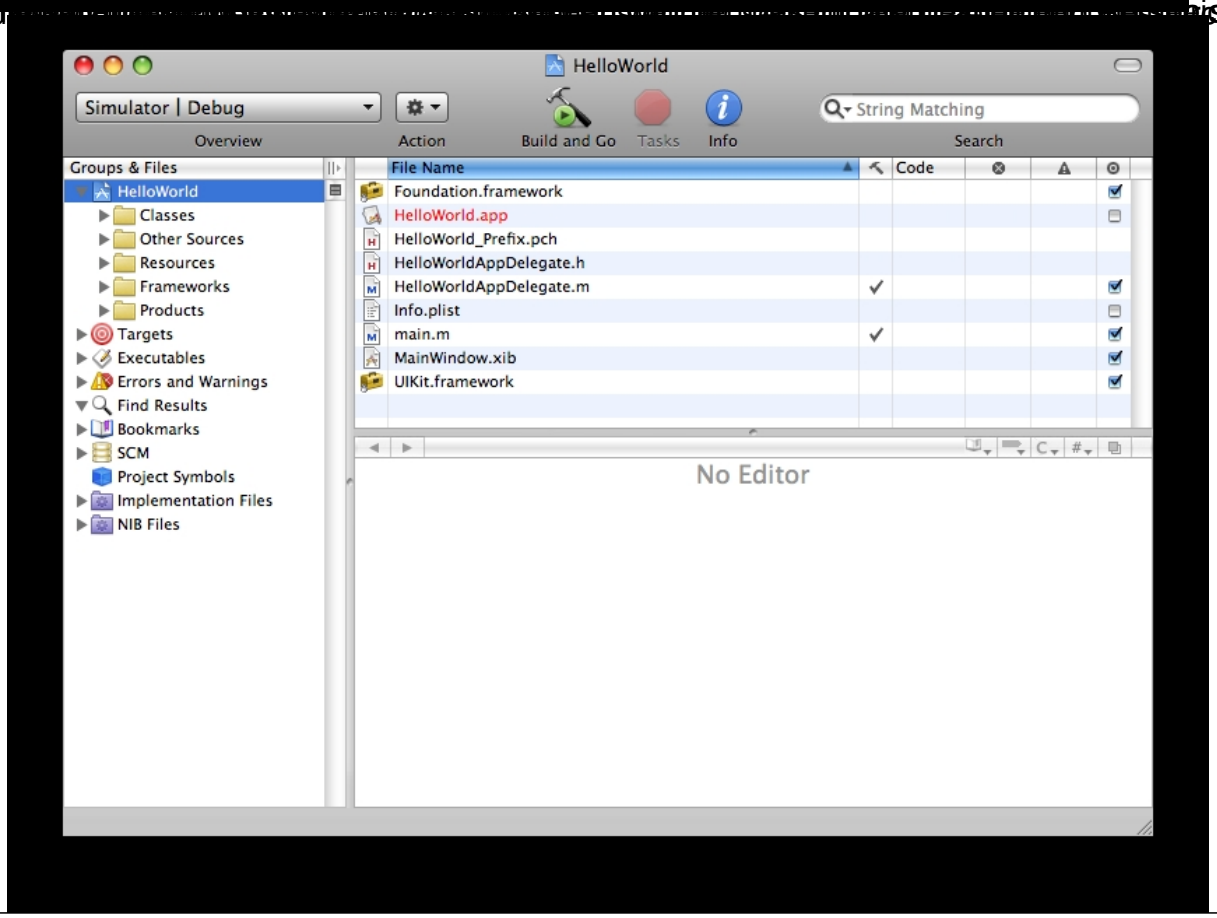
Wir wollen nun erst einmal das benötigte Projekt erstellen. Wählt dazu aus dem Hauptmenü von Xcode den Eintrag "File->New Project ..."

Es öffnet sich ein Dialog, in welchem Ihr die Art des Projektes auswählen müsst:



"[illegible]" Gebt hier


Save As: HelloWorld
Development
search
DEVICES
Macintosh ...
Untitled
iDisk
MoveMe
SHARED
PLACES
Schreibtisch
philipp.ho...
Programme
Dokumente
Bilder
Dev
Development
Dokumente
Downloads
Filme
Library
mpowerplay
Musik
netbeanstest
Öffentlich
Schreibtisch
New Folder
Cancel
Save


HelloWorld
Simulator | Debug
Overview
Action
Build and Go
Tasks
Info
String Matching
Search
Groups & Files
HelloWorld
Classes
Other Sources
Resources
Frameworks
Products
Targets
Executables
Errors and Warnings
Find Results
Bookmarks
SCM
Project Symbols
Implementation Files
NIB Files
File Name
Code
Foundation.framework
HelloWorld.app
HelloWorld_Prefix.pch
HelloWorldAppDelegate.h
HelloWorldAppDelegate.m
Info.plist
main.m
MainWindow.xib
UIKit.framework
No Editor

2. Hello World realisieren

So, wir haben jetzt also unser "HelloWorld" Projekt mit allen benötigten Dateien und Resourcen erstellt. Theoretisch könntet Ihr diese Applikation jetzt schon auf dem iPhone-Simulator starten (cmd + Enter), würdet jedoch nichts als einen leeren Bildschirm auf dem iPhone sehen. Jetzt werden wir den Text "HelloWorld" hinzufügen.

Öffnet bitte die Datei HelloWorldAppDelegate.m - diese ist zuständig für die Darstellung des Start-Bildschirms der Hello World Anwendung. Mit Start-Bildschirm meine ich an dieser Stelle ein Window. Jede iPhone-Applikation hat genau ein Window. Möchte man später mehrere Ansichten in seiner Applikation hinzufügen, so wird man für jede dieser Ansichten eine View zu diesem Window hinzufügen. In unserem Beispiel reicht jedoch ersteinmal nur das Window auf welchem wir jetzt eine Navigationsleiste darstellen werden und diese mit dem Titel "Hello World" ausstatten.

Wenn Ihr die Datei HelloWorldAppDelegate.m geöffnet habt, findet Ihr in dieser die Funktion a pplicationDidFinishLaunching
. Diese wird zum Start des Programmes aufgerufen und ist für das zeichnen des Windows verantwortlich. Wir erweitern jetzt diese Funktion um die Navigationsleiste. So sollte Eure Funktion im Moment noch aussehen:

```
- (void)applicationDidFinishLaunching:(UIApplication *)application {

    // Override point for customization after app launch
    [window makeKeyAndVisible];
}
```

Erweitert diese nun wie gefolgt gezeigt (Änderungen sind **fett** dargestellt):

```
- (void)applicationDidFinishLaunching:(UIApplication *)application {

    // Override point for customization after app launch

    // die Navigationsleiste
    UINavigationBar *navBar = [[UINavigationBar alloc] initWithFrame:
 □□ □□□□ □□□ □□□ □□□ □□□ □□□ □□ CGRectMake( 0.0f, 20.0f, window.frame.size.width,
48.0f)];

 □□□ // Aussehen der Navigationsleiste
 □□ □[navBar setBarStyle: 0];
 □□ □
 □□ □// der Titel
 □□ □UINavigationItem *title = [[UINavigationItem alloc] initWithTitle:@"Hello World"];
 □□ □
 □□□ // Titel wird der Navigationsleiste hinzugefügt
 □□□ [navBar pushNavigationItem:title animated:true];
```

```
    // die navigationsleiste wird dem window hinzugefügt
    [window addSubview:navBar];

    [window makeKeyAndVisible];

    // Speicher freigeben
    [title release];
    [navBar release];
}
```

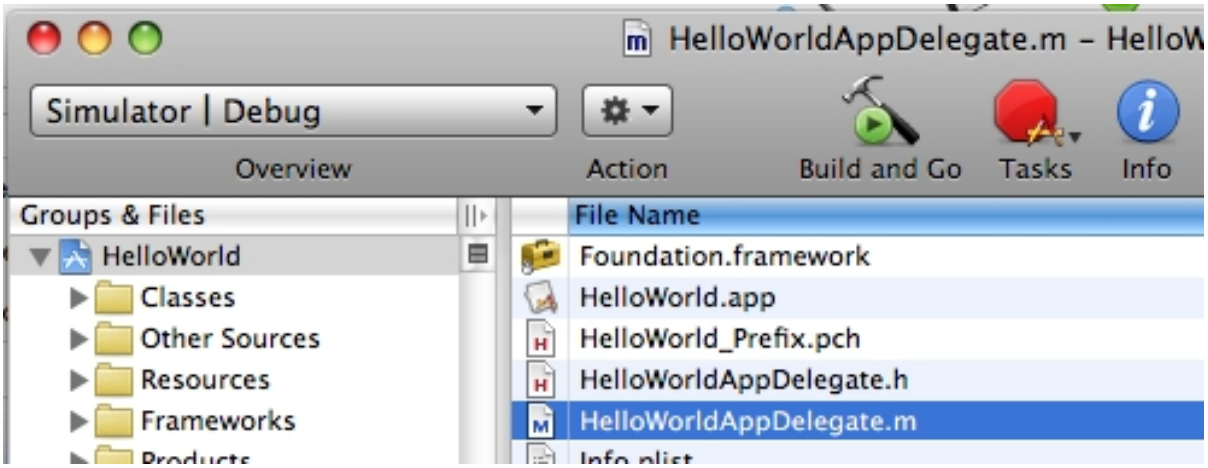
Das war es auch schon. Wir erstellen ein Objekt vom Typ UINavigationBar und zusätzlich eins vom Typ UINavigationItem .
Dieses fügen wir dem NavigationBar Objekt hinzu und packen alles auf unser Window. Zum Schluß geben wir noch den Speicher wieder frei, welchen unsere Navigationsleiste beansprucht hat (
[title release][navBar release];
).

Im nächsten Abschnitt wird gezeigt wie die neue Applikation gestartet wird.

---

3. Bereitstellen und Starten

Jetzt brauchen wir nur noch das Programm auf den Simulator zu installieren und starten - das erledigt Xcode für uns wenn wir auf "Build and Go" klicken wie hier gezeigt.



Und nun - siehe da, startet der iPhone-Simulator auch gleich - wir sehen kurz das SpringBoard (Hauptmenü) und unsere Applikation wird gestartet.

oWorld.zip