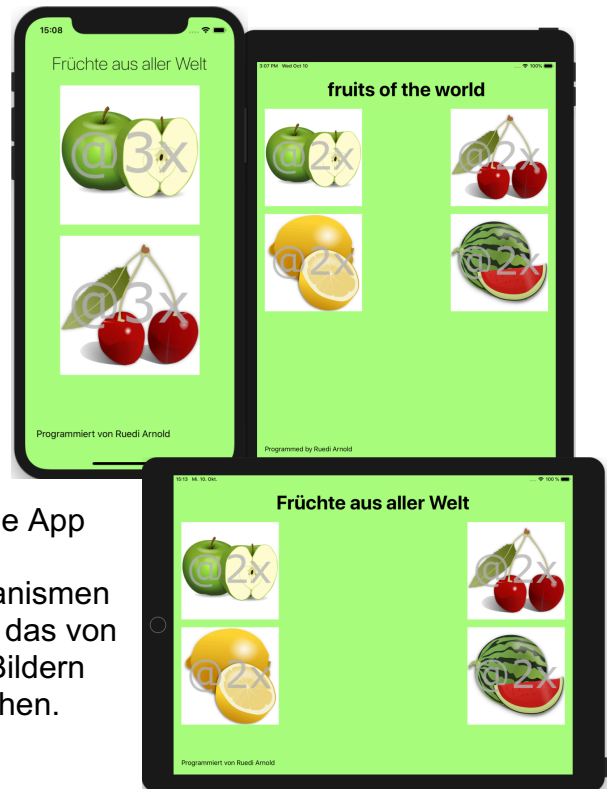


## Übung 5: Adaptives Layout, @2x/@3x-Bilder und Mehrsprachigkeit

Diese Übung hat drei inhaltliche Schwerpunkte. Einerseits werden sie mithilfe von Size Classes eine App mit adaptivem Layout entwickeln. Ziel ist es die Benutzer-Oberfläche für ein aktuelles iPad (Hoch- und Querformat) sowie ein aktuelles iPhone (nur Hochformat) zu optimieren. Andererseits wird die App mehrsprachig entwickelt, sie werden diese also lokalisieren und die entsprechenden iOS-Mechanismen kennen lernen. Im Weiteren verwenden wir hier das von Apple zur Verfügung gestellte Konzept um mit Bildern für hochauflösende Retina-Bildschirme umzugehen.

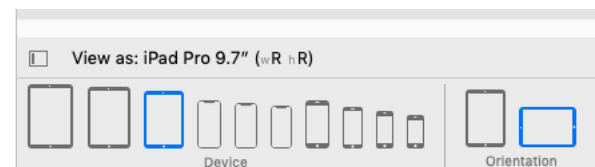


### 1. Neues Projekt "AdaptiveApp"

Erstellen sie in Xcode ein neues Projekt, wählen sie "Single View Application" und dann Projektname "AdaptiveApp" und keine Verwendung von Core Data, Unit Tests usw. Nun können wir mit der Entwicklung der universellen Oberfläche (Size Class "w Any h Any") loslegen. Färben sie die View beliebig ein, sodass sie später die Abstände zwischen den Bildern besser erkennen.

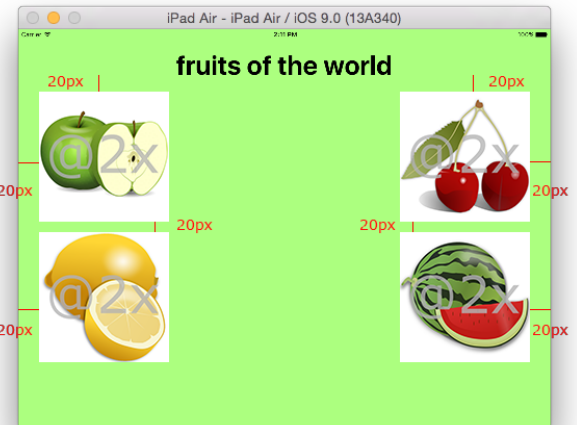
### 2. Applikationstitel zeichnen im Interface Builder

Stellen sie im Interface Builder unter "View as:" einen iPad-Bildschirm im Querformat ein, wir arbeiten also in der Size Class (wR hR), siehe Screenshot rechts. Ziehen sie aus der "Object Library" ein Label auf den View Controller im Storyboard. Schreiben sie den Titel "fruits of the world" hin. Geben sie dem Label mithilfe des "Attribute Inspector" die Schriftgrösse 50 sowie den Style bold. Nun möchten wir, dass der Titel unabhängig von der Bildschirmgrösse immer horizontal zentriert ist. Wählen sie dazu den alignButton in der "auto layout toolbar" (unten im Interface Builder) aus und aktivieren die Constraint "Horizontally in Container" und setzen dessen Wert auf 0. Mit einem Klick auf "Add Constraints" tritt das Constraint in Kraft. Weiter soll der Titel auf allen Geräten 20px Abstand von der oberen "Safe Area" haben. Wählen sie dazu den pin-Button in der "auto layout toolbar" aus, definieren das korrekte Constraint und fügen diese mit "Add Constraints" hinzu. Danach sollte das Titel-Label die rechts im Screenshot angegebenen Constraints haben.



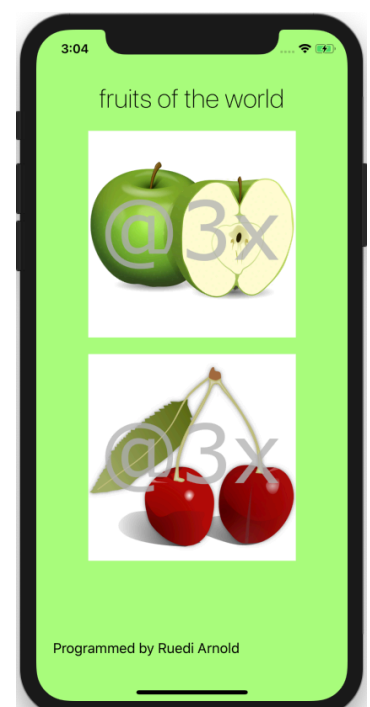
### 3. Bilder hinzufügen und Constraints für die universelle Oberfläche definieren

Fügen sie dem Projekt die benötigten vier Bilder hinzu. Verwenden sie dazu den in der Vorlesung kurz gezeigten Asset-Mechanismus. Um die Bilder auf dem Storyboard anzuzeigen, verwenden wir jeweils eine UIImageView aus der "Object Library". Einer UIImageView kann mittels "Attribute Inspector" das Bild zugeordnet werden. Platzieren sie die vier Image Views ungefähr gemäss nebenstehenden Printscreens und weisen sie die entsprechenden Bilder zu. Insbesondere sollen auf dem iPad also die **Kirschen rechts vom Apfel** sein. Damit nun die Bilder auf allen Geräten wie gewünscht dargestellt werden, müssen als nächstes Constraints definiert werden. Implementieren sie die Constraints gemäss den im Screenshots rechts oben **in rot angegebenen Werten**, sprich an den angegebenen Orten jeweils einen Abstand von 20px. Testen sie nun ihre App jeweils im Simulator für ein iPhone und ein iPad. Sie werden feststellen, dass das adaptive Layout für iPads bereits zufriedenstellend wie ist und die korrekten Bilder dargestellt werden (sprich je nach Gerät und Auflösung die passenden @2x- bzw. @3x-Bilder). Doch das Layout ist für ein iPhone nicht optimal. Der Titel ist zu gross und zudem werden die Bilder überlappend dargestellt, siehe Screenshot rechts.



### 4. Oberfläche für ein iPhone im Hochformat optimieren

In einem ersten Schritt möchten wir die Darstellung der Bilder für iPhone-Bildschirme optimieren. Aus Platzgründen sollen auf dem iPhone nur der **Apfel und darunter die Kirsche** angezeigt werden, und zwar mit Abstand 20px. Um ein Element in einer bestimmten Size Class auszublenden, muss die Checkbox "Installed" im "Attribute Inspector" einer bestimmten Size Class deaktiviert werden. Hinweis: Die Size Class "w Compact h Regular" wird für das iPhone im Hochformat verwendet. Im weiteren Schritt müssen die Constraints für die übrigen Bilder angepasst werden, so dass diese nicht, bzw. am gewünschten Ort angezeigt werden. Wechseln sie dazu mithilfe der "auto layout toolbar" in die passende Size Class, löschen sie für diese Size Class alle Constraints im "Size inspector" des Bildes und definieren sie neue Constraints, sodass der Screen gemäss den Anforderungen aussieht. Nun muss noch der Titel für das iPhone im Hochformat optimiert werden. Dazu wählt man das Label an und wechselt in den "Attributes Inspector". Dort können mithilfe des "+" weitere Eigenschaften der Schrift für eine bestimmte Size Class definiert werden. Wählen sie für den Titel auf einem iPhone im Hochformat die Schriftgrösse 32 sowie den Style thin. Das sollte dann auf einem iPhone 7 Plus aussehen wie auf dem Screenshot rechts.



## 5. Mehrsprachigkeit

Richten sie das Projekt für Mehrsprachigkeit ein. Vorgesehen ist mindestens eine Deutsche und Englische Version vom Titel, lokalisiert in der Storyboard-Datei. Erstellen sie die benötigten Dateien und fügen sie diese ihrem Projekt hinzu. Verwenden sie den in der Vorlesung gezeigten "Export For Localizations... / Import Localizations" Mechanismus. Wechseln sie die Sprache des iOS und testen sie im Simulator ob nun der Titel passend in Englisch und Deutsch angezeigt wird. Nun möchten wir noch einen Text mehrsprachig haben, welcher durch ein IBOutlet auf ein UILabel aus dem Code gesetzt wird. Fügen sie dazu der App links unten ein UILabel mit folgenden Constraints hinzu: Abstand vom linken und vom unteren Rand je 20px, so dass es aussieht wie in den beiden Screenshot unten. Als letzter Schritt realisieren sie die Mehrsprachigkeit des Labels mithilfe des NSLocalizedString-Mechanismus, d.h. also inklusive von Dateien mit dem Namen Localizable.strings in den entsprechenden lproj-Verzeichnissen.

