

Full Stack iOS Entwicklung mit Swift

WPF im MIM - SS 17
Alexander Dobrynin M.Sc.

Heute

Autolayout
Size-Classes

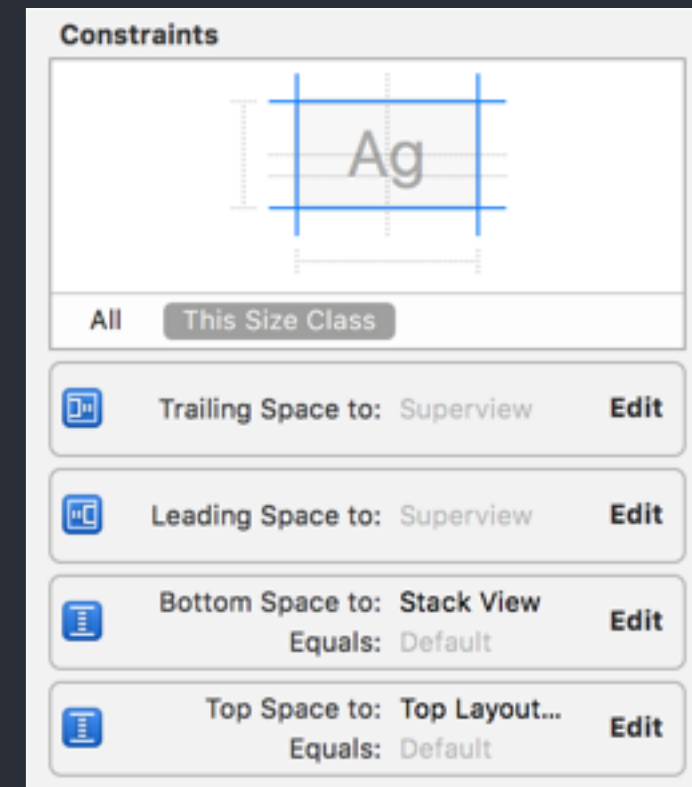
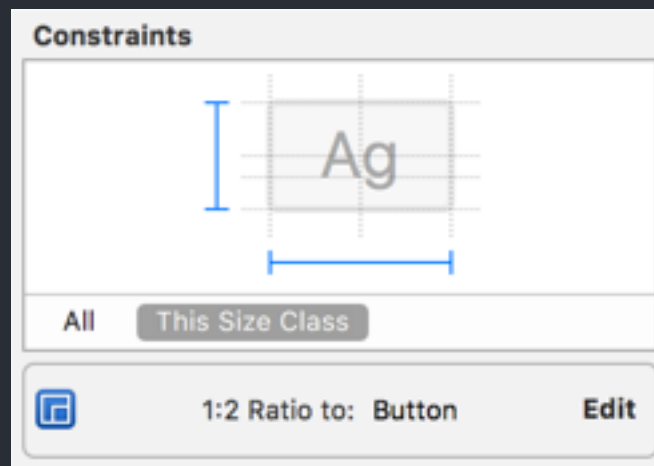
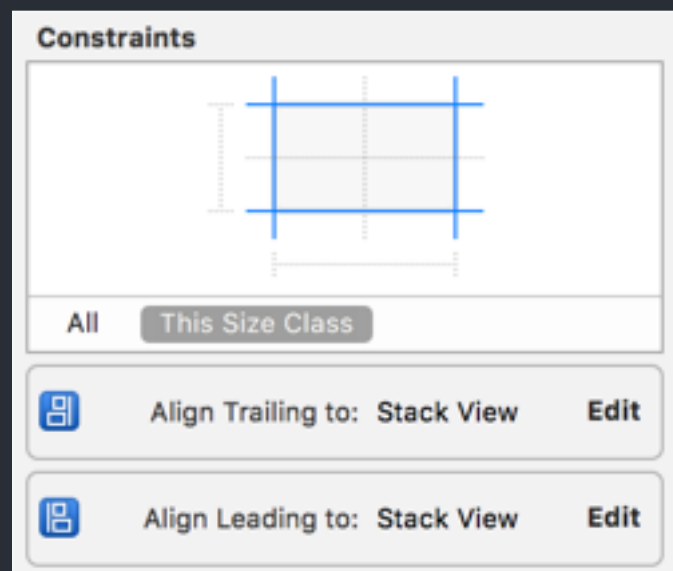
Demo - MatchingCardGame

Autolayout

- Setzen von **Constraints** für Views
- Ausrichtung von Views an den **blauen Linien**
 - entweder **Reset to Suggested Constraints**
 - oder **Crtl-Drag** von View zu View oder von View zu Kanten (Edges)
- Verifizieren im **Size Inspector**
- Eliminieren von **Magic Numbers** - setzen auf **Default** oder **0**
- Konflikte auflösen in der **Document Outline**

Autolayout

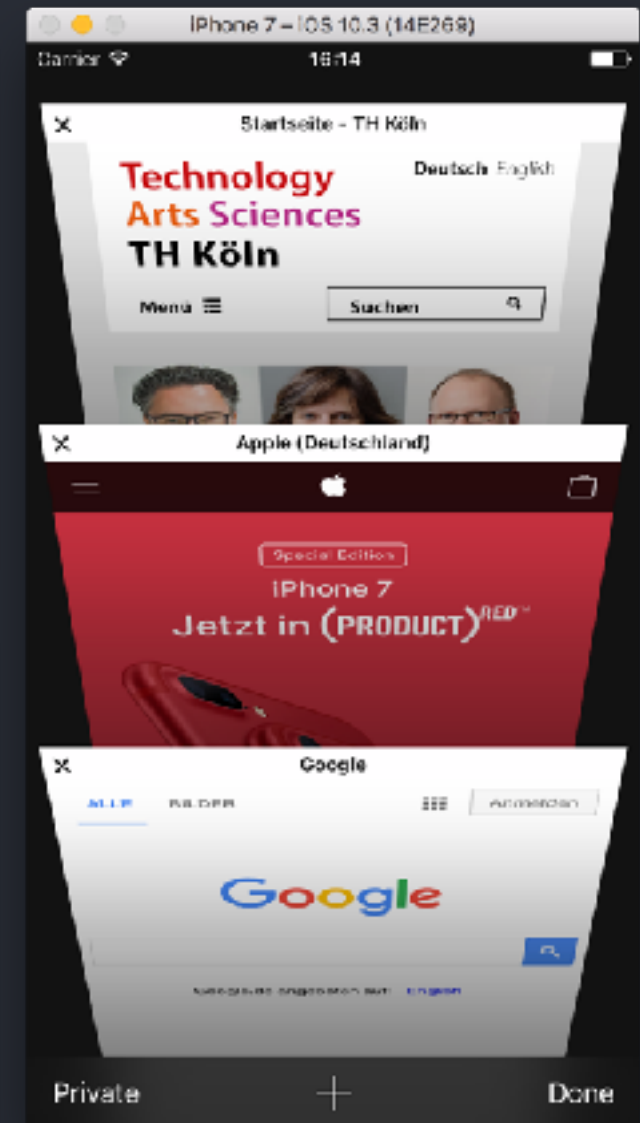
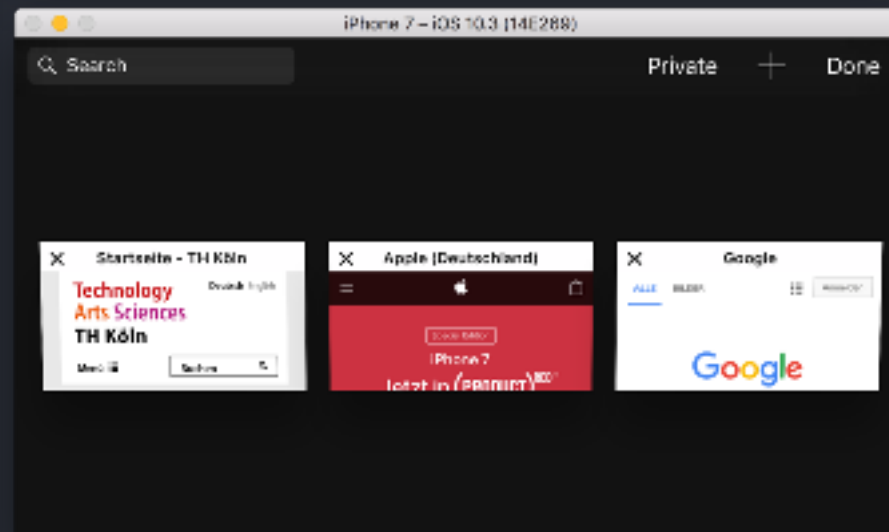
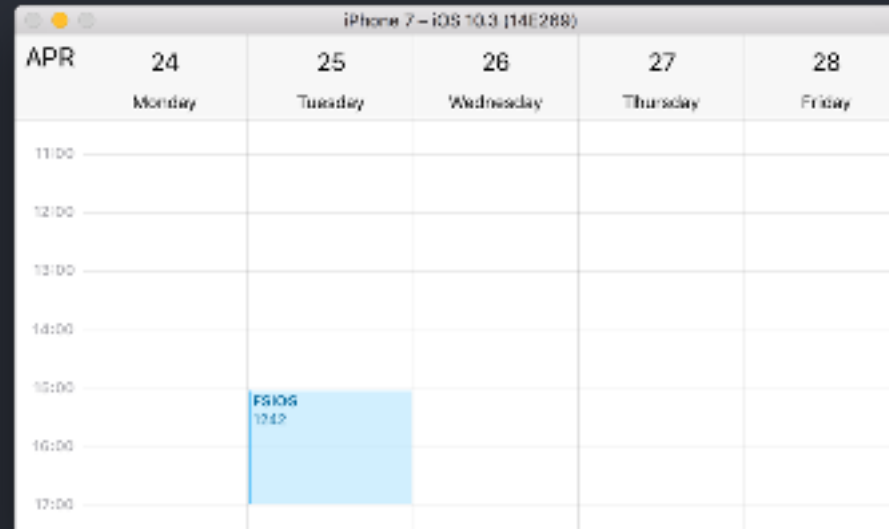
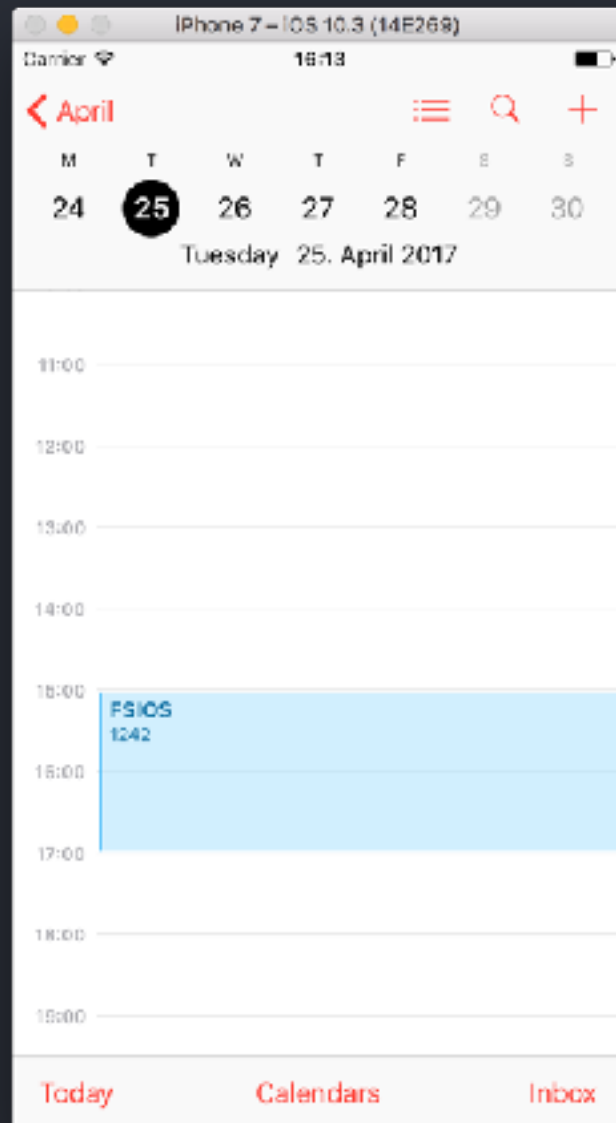
- Constraints sind
 - Leading-, Trailing-, Top- oder Bottom- Spaces, relativ zu einer View und einem Offset von n, 0 oder Default
 - Leading-, Trailing-, Top- oder Bottom- Alignments zu einer View
 - Center Vertically/Horizontally zu einer View
 - fixe Werte für Width, Height oder Aspect-Ratio



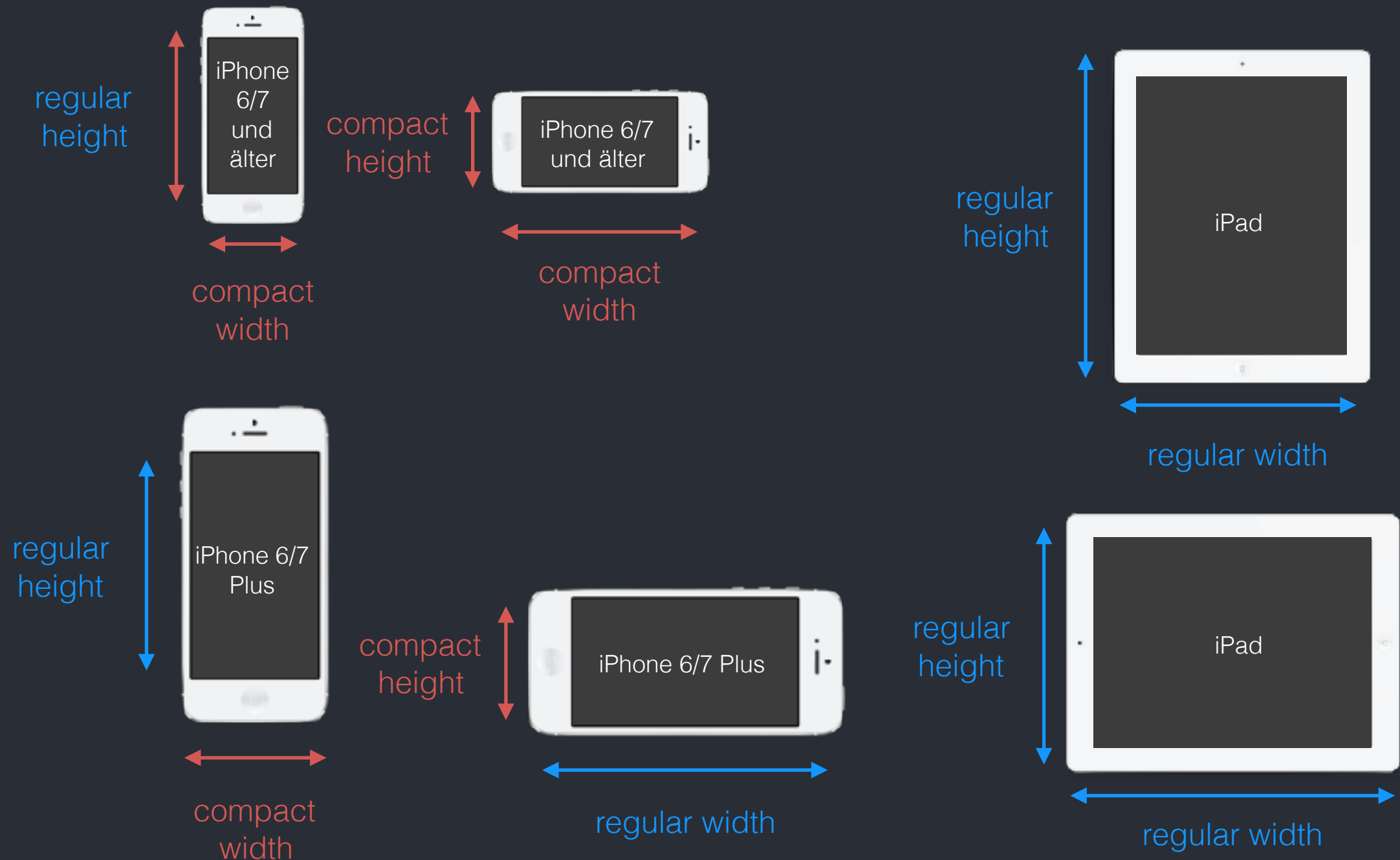
Autolayout

- Universal Apps
- Funktionierende Views für unterschiedliche Displaygrößen und Auflösungen
- Rotationen (Landscape und Portrait)
- ViewController für iPhone und iPad
- Size Classes - Compact und Regular

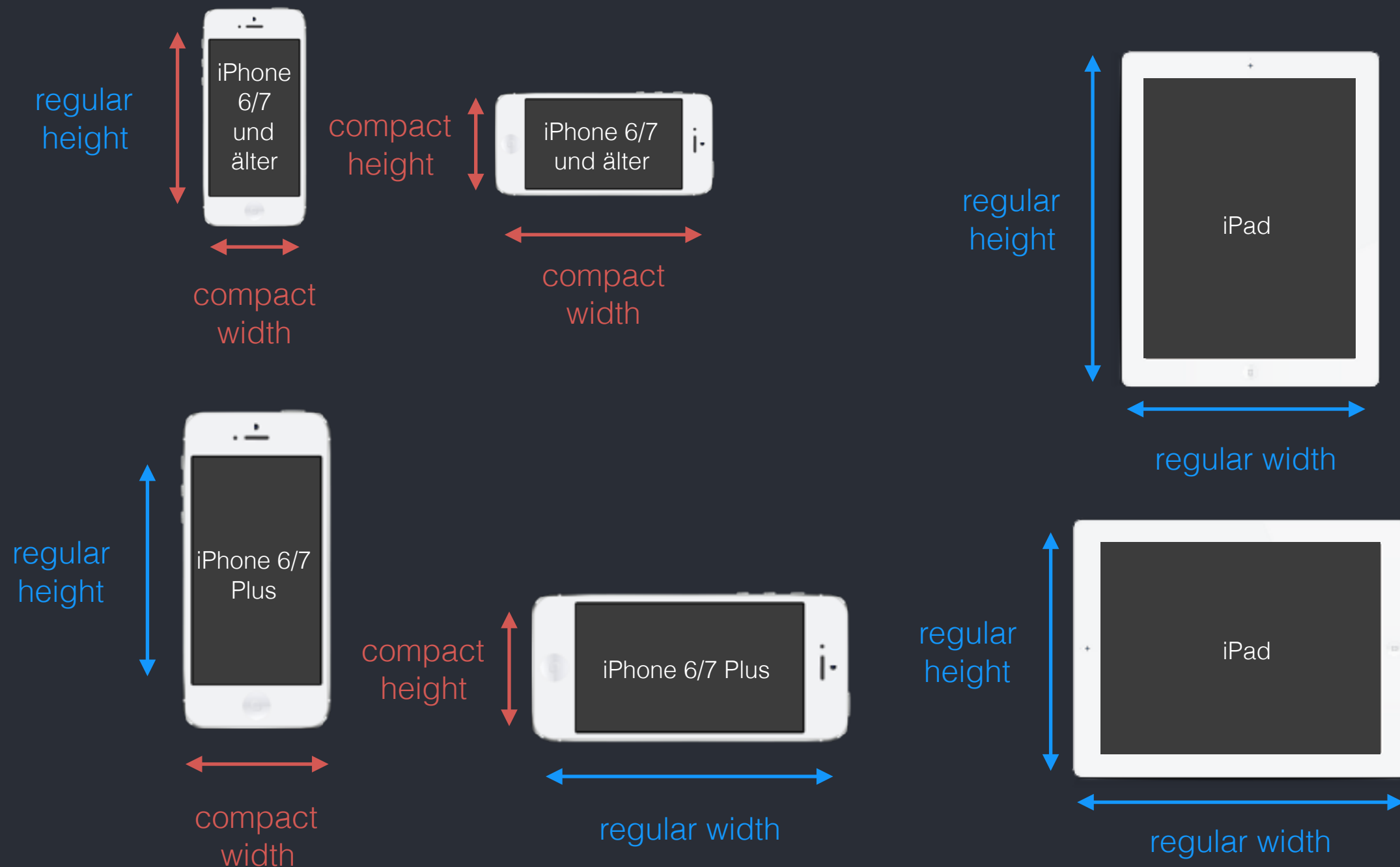
Autolayout



Autolayout



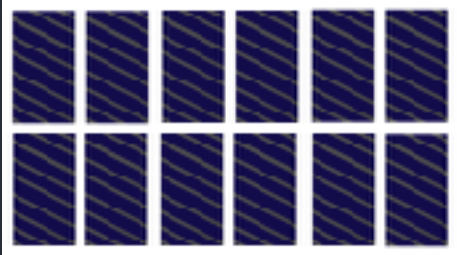

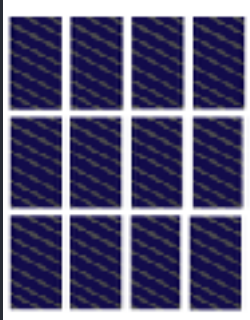
Autolayout



Autolayout



Autolayout

	compact width	any width	regular width
compact height			
any height			
regular height			

Nächste Woche

Start des Hackathon 🧐🎉
Swift und Foundation

Demo - MatchingCardGame

Custom View
Autolayout
Size-Classes