Mastering Auto Layout

About me

Libranner L. Santos Espinal

Software Engineer
Author/ Tech Editor @ Ray Wenderlich
Swift/ iOS Enthusiast

Follow me on Twitter: **@libranner**

Mastering Auto Layout: Outline

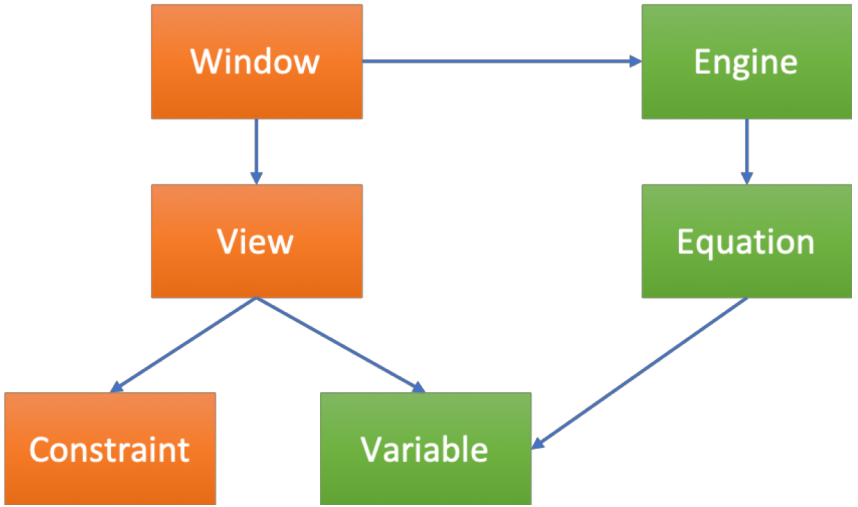
- Important Concepts Before We Start
- Debugging
  - Getting to know the Sample App
  - Types of Errors
  - Tools and Techniques
- Code vs Xibs vs Interface Builder
- Tips and Tricks
  - General
  - Interface Builder
  - Performance
- Sum it up
- Where to go from here?

Important Concepts Before We Start

- The Auto Layout Engine
- Intrinsic Content Size & Alignment Rectangles
- Render Loop

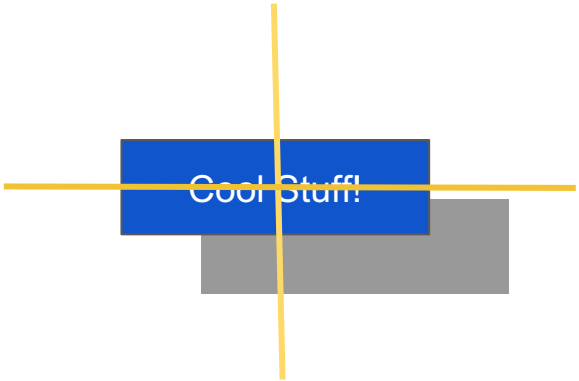
Important Concepts Before We Start

- **The Auto Layout Engine**
- Intrinsic Content Size & Alignment Rectangles
- Render Loop



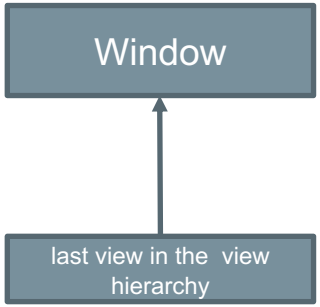
Important Concepts Before We Start

- The Auto Layout Engine
- **Intrinsic Content Size & Alignment Rectangles**
- Render Loop



Important Concepts Before We Start

- The Auto Layout Engine
- Intrinsic Content Size & Alignment Rectangles
- **Render Loop**



Debugging

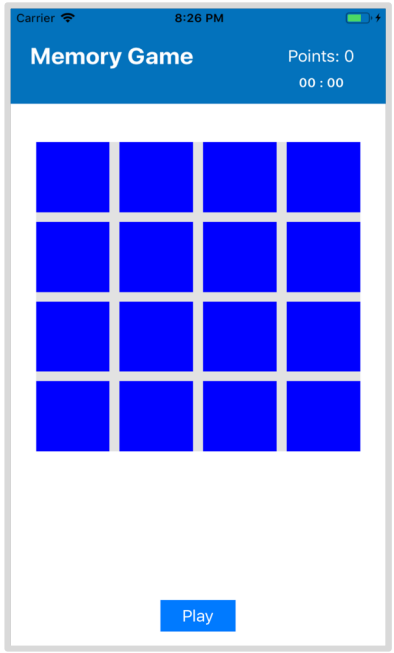
Type of Errors

- Logical Errors
- Unsatisfiable Constraints
- Ambiguous Constraints
  - Content Hugging
  - Content Compression Resistance

… And :

- Performance

Getting to Know the Sample App



Tools and Techniques

- Visual Format Language. Why is necessary to talk about VFL?

H:|-[icon(==iconDate)]-20-[iconLabel(120@250)]-20@750-[iconDate(>=50)]-|

Source: https://www.raywenderlich.com/277-auto-layout-visual-format-language-tutorial

Demo

Tools and Techniques

- Understanding the Logs
- Symbolic Breakpoints

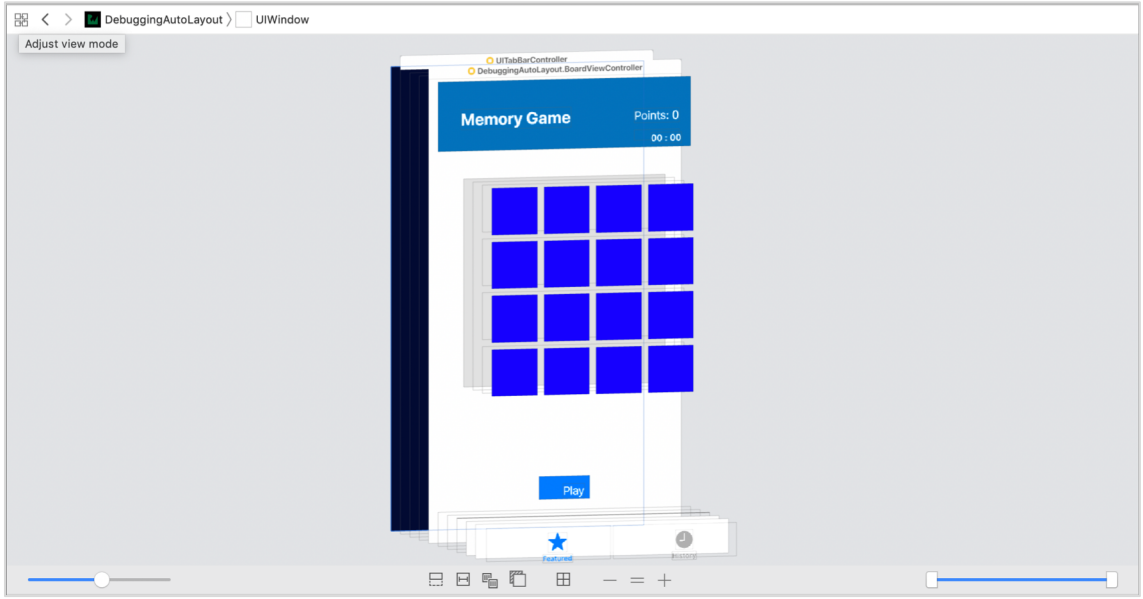
```
2019-05-22 11:48:06.901162+0300 DebuggingAutoLayout[21302:999431] [LayoutConstraints] Unable to simultaneously
  satisfy constraints.
    Probably at least one of the constraints in the following list is one you don't want.
    Try this:
        (1) look at each constraint and try to figure out which you don't expect;
        (2) find the code that added the unwanted constraint or constraints and fix it.
(
    "<NSLayoutConstraint:0x60000021f1b0 HeaderView.top ==
      UILayoutGuide:0x600001820460'UIViewSafeAreaLayoutGuide'.top   (active, names: HeaderView:0x7fced3106ee0
      )>",
    "<NSLayoutConstraint:0x60000021f0c0 'HeaderBottom'
      UILayoutGuide:0x600001820460'UIViewSafeAreaLayoutGuide'.bottom == HeaderView.bottom + 600   (active,
      names: HeaderView:0x7fced3106ee0 )>",
    "<NSLayoutConstraint:0x60000021ff20 'UIView-Encapsulated-Layout-Height' MainView.height == 667   (active,
      names: MainView:0x7fced31081f0 )>",
    "<NSLayoutConstraint:0x60000021f4d0 'UIViewSafeAreaLayoutGuide-bottom'
      V:[UILayoutGuide:0x600001820460'UIViewSafeAreaLayoutGuide']-(49)-|   (active, names:
      MainView:0x7fced31081f0, '|':MainView:0x7fced31081f0 )>",
    "<NSLayoutConstraint:0x60000021f3e0 'UIViewSafeAreaLayoutGuide-top'
      V:|-(20)-[UILayoutGuide:0x600001820460'UIViewSafeAreaLayoutGuide']   (active, names:
      MainView:0x7fced31081f0, '|':MainView:0x7fced31081f0 )>"
)

Will attempt to recover by breaking constraint
<NSLayoutConstraint:0x60000021f0c0 'HeaderBottom'
  UILayoutGuide:0x600001820460'UIViewSafeAreaLayoutGuide'.bottom == HeaderView.bottom + 600   (active, names:
  HeaderView:0x7fced3106ee0 )>

Make a symbolic breakpoint at UIViewAlertForUnsatisfiableConstraints to catch this in the debugger.
The methods in the UIConstraintBasedLayoutDebugging category on UIView listed in <UIKitCore/UIView.h> may also
  be helpful.
```

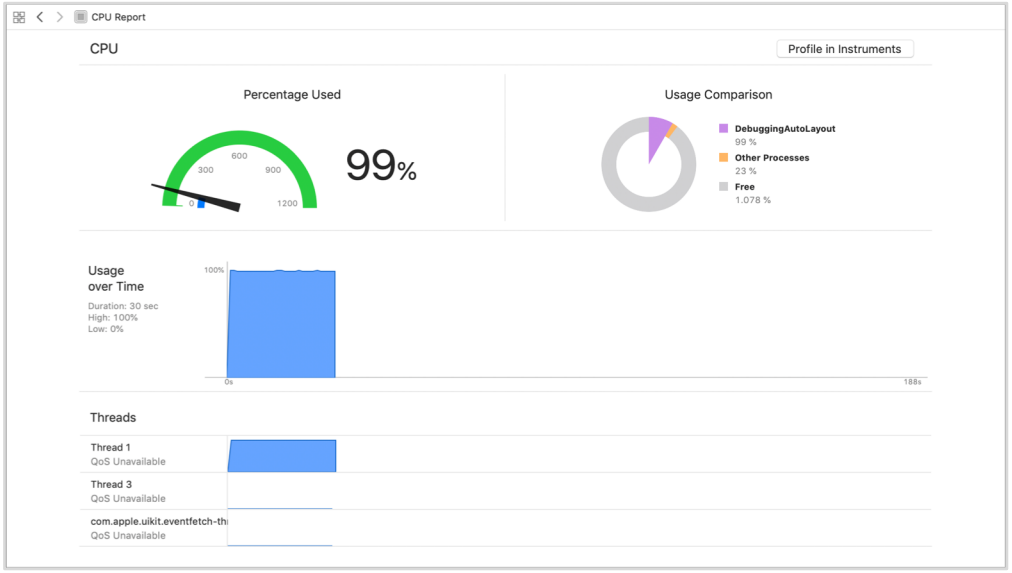
Tools and Techniques

- View Debugging
  - View Hierarchy



Perfomance Issues

- Churning
- Auto Layout Feedback Loop



Code vs Xibs vs Interface Builder

- What's the right choice?

Team

Complexity

How important is reusability?

Tips & Tricks

Tips and Tricks: General

- wtfautolayout.com
- Use the view methods
  - hasAmbiguosLayout
  - constraintsAffectingLayout
  - exerciseAmbiguityInLayout

Tips and Tricks: General

- Be aware of Layout Guides
- Use StackViews!
- Use View Containers

Tips and Tricks: Interface Builder

- Use the Preview Window
- Identify your constraints. Your future self will appreciate it.
- Use resizable and scalable images. When possible slice them up.
- Don’t forget you can use IBInspectable / IBDesignable :]

Tips and Tricks: Performance

- Don’t create layouts that try to accomplish a LOT of scenarios.
- Inequalities are cheap, use them :]
- Constraint Constants, are managed by the Engine, and since the engine is a dependency tracker, it won’t cost much.
- Try to use priorities wisely. Error minimization has a significant cost.

Tips and Tricks: Performance II

- Avoid churning, make sure you update the constraints only if it's necessary.
  - Avoid removing constraints unless it's necessary
  - When possible hide views instead of removing them. (StackViews for the win!)
- Overriding the intrinsicContentSize property can help with performance.
- systemLayoutSizeFitting is expensive, try to avoid it.

Sum it up

Where to go from here?

- Learn Auto Layout:
  - https://www.raywenderlich.com/7478-beginning-auto-layout
  - https://www.raywenderlich.com/4260-mastering-auto-layout
  - https://www.raywenderlich.com/277-auto-layout-visual-format-language-tutorial
- Best practices:
  - https://www.raywenderlich.com/5160-auto-layout-best-practices
- Performance:
  - https://developer.apple.com/videos/play/wwdc2018/220

спасибо за ваше внимание

About me

Libranner L. Santos Espinal

Software Engineer
Author/ Tech Editor @ Ray Wenderlich
Swift/ iOS Enthusiast

Follow me on Twitter: **@libranner**