

FROM DESIGN TO code

NSSpain 2016 • @PUMPKIN + @GONZALEZREAL

**DESIGNER + DEVELOPER
WITH A SMALL SIDE PROJECT**

- PART ONE -



DESIGNING THE RIGHT PRODUCT THE RIGHT WAY

First

OUR, NOT MINE

MY APP

- ▶ I had this great idea
- ▶ I designed these awesome mockups
- ▶ You code it for me

**GREAT
GREAT
INCOMPLETE
FAIL**



MY APP?

OUR APP?

- ▶ What do you think?
- ▶ How would you make it better?
- ▶ Do you want to do this together?

OUR APP!



SOLID
COMPLETE
LEARN SOMETHING

Second

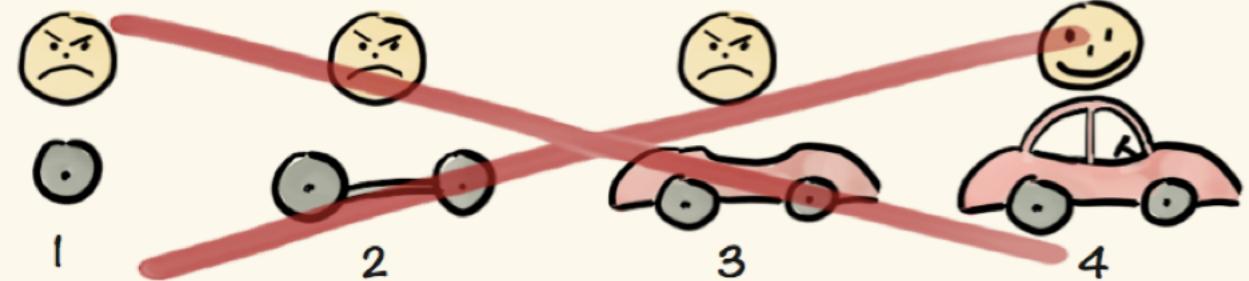
A BATTLE FOR RESOURCES

TIME

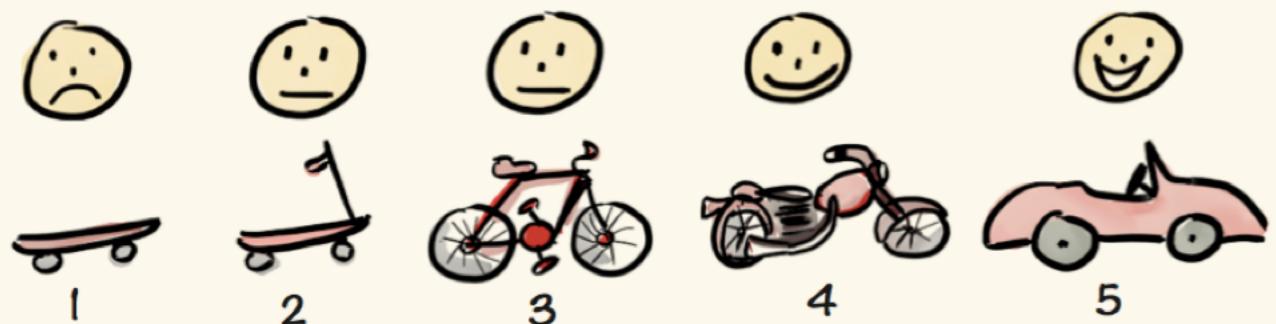
and

HUMAN

Not like this....



Like this!



Henrik Kniberg

CUT THE SCOPE
DON'T REINVENT THE WHEEL
REUSE
AND ITERATE

blog.crisp.se/2016/01/25/henrikkniberg/making-sense-of-mvp

Third

DESIGN SYSTEMS

**TO BREAK THE RULES
YOU NEED TO KNOW THEM**

HUMAN INTERFACE GUIDELINES

developer.apple.com/design

Aesthetic Integrity

Consistency

Direct Manipulation

Feedback

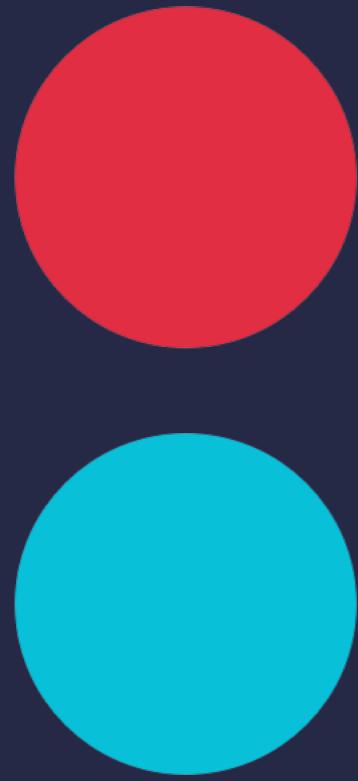
Metaphors

User Control

STAY TRUE TO THE PLATFORM

Aa

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
1 2 3 4 5 6 7 8 9 0 & @ . , ? ! ' " ()

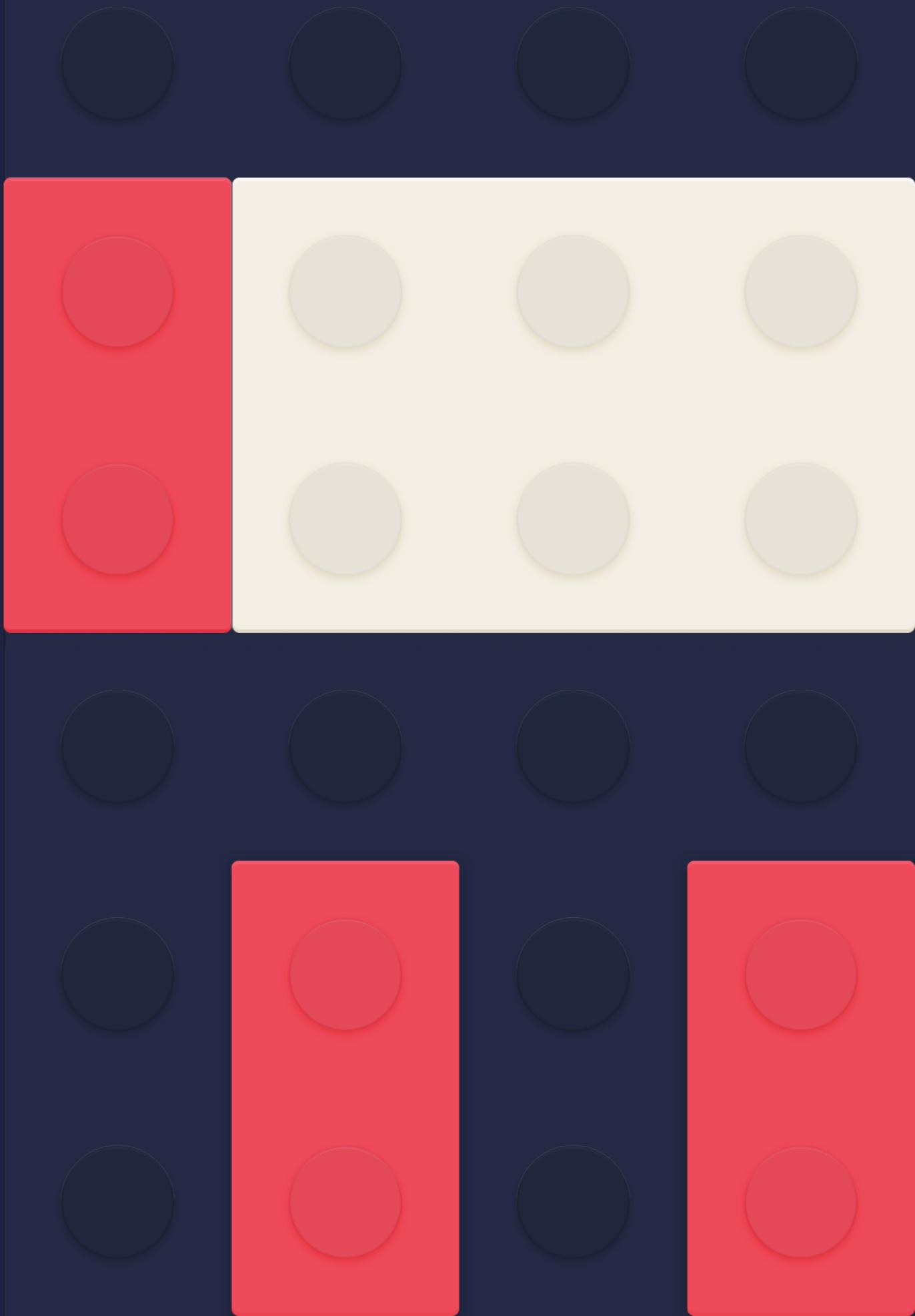


WCAG 2.0

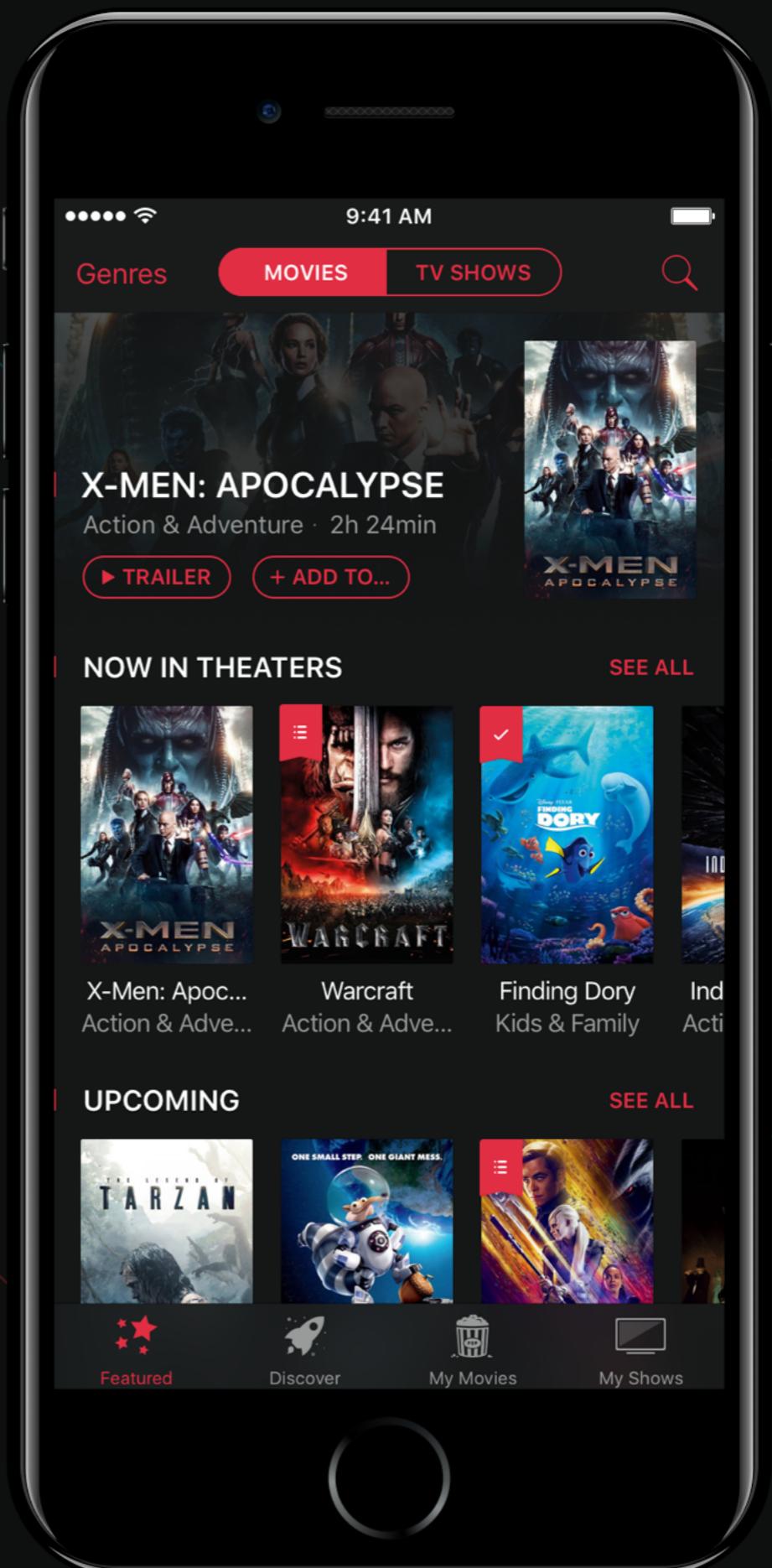
TYPOGRAPHY AND COLOR

lamb.cc/typograph · leaverou.github.io/contrast-ratio

BUILD A SYSTEM OF REUSABLE COMPONENTS



OUR APP



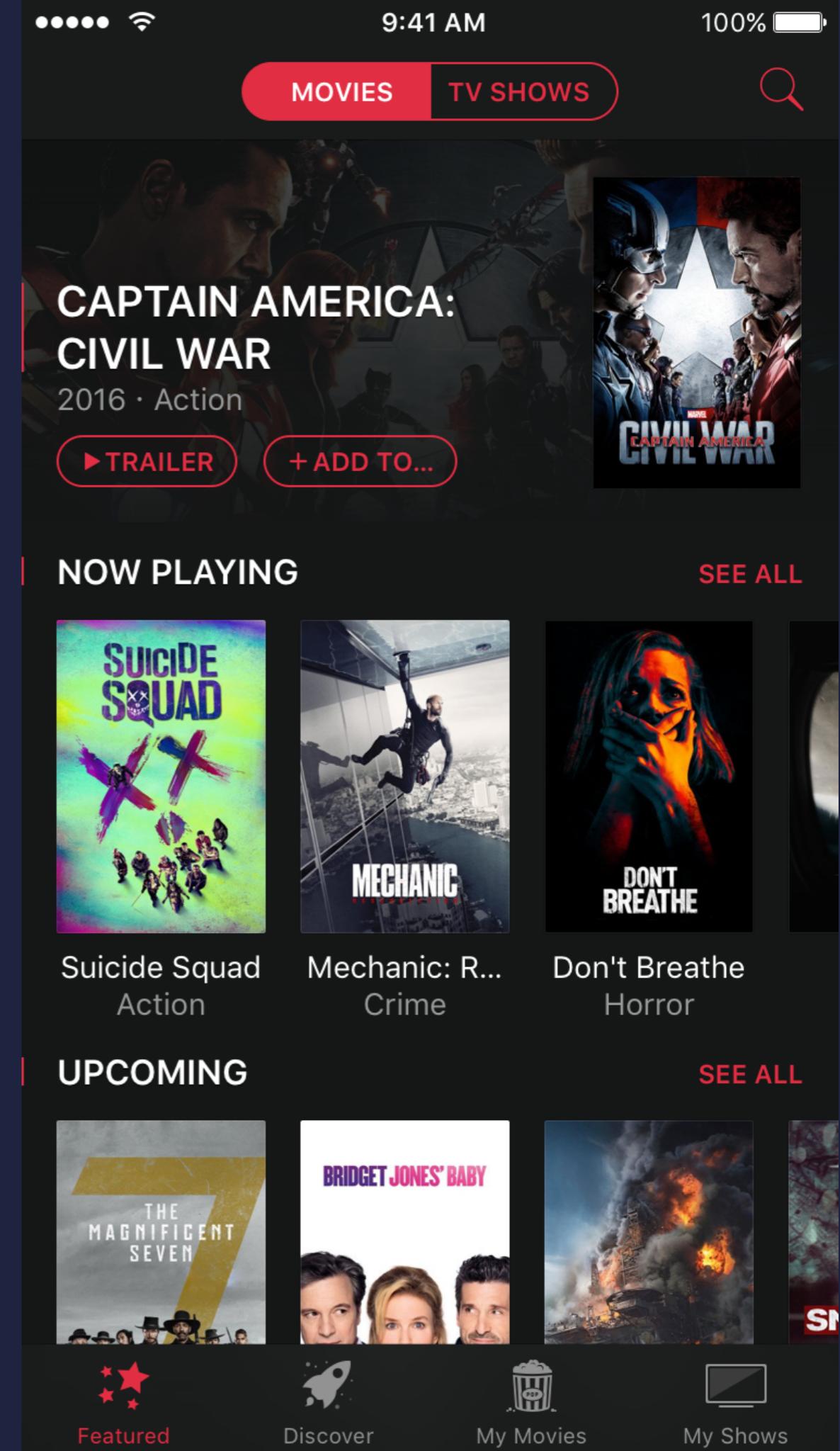
- PART TWO -

BUILDING UI COMPONENTS IN SWIFT



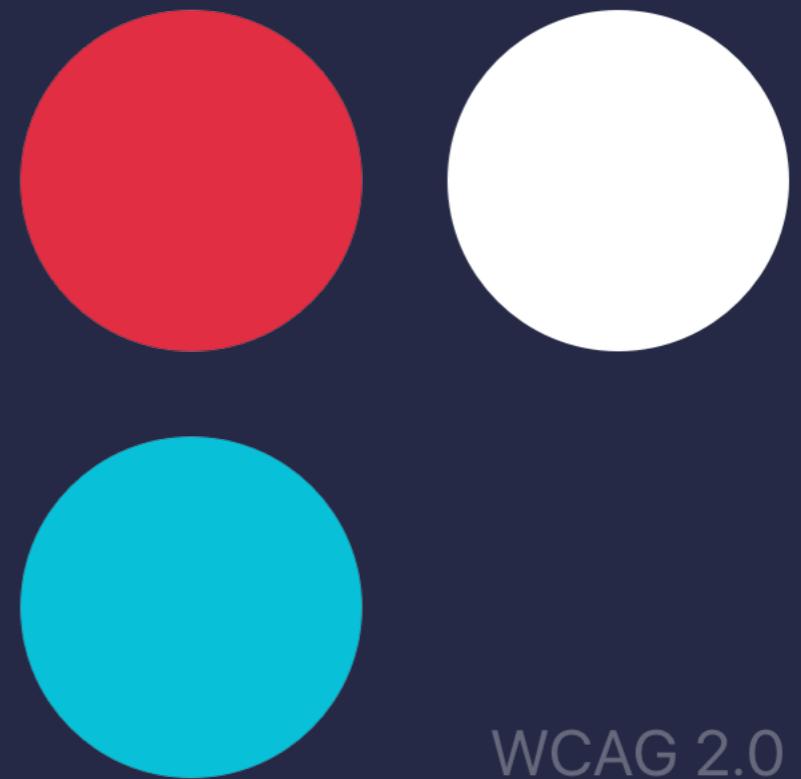
UI COMPONENTS

- ▶ Button, Label, PosterView, BackdropView
- ▶ PosterCell, CardCell
- ▶ ViewControllers!?



First

COLORS



WCAG 2.0

DON'T

```
view.backgroundColor = UIColor(red: 0.09,  
                                green: 0.08,  
                                blue: 0.08,  
                                alpha: 1.0)
```

INSTEAD

- ▶ Name your colors
- ▶ DarkGray, OliveGreen
- ▶ Background, Text, Action
- ▶ ❤️ SwiftGen

github.com/AliSoftware/SwiftGen

USING SWIFTGEN

Colors.txt

Movie: #E22E43

Show: #09C0D9

Background: #151717

...

```
$ swiftgen colors Colors.txt --output Color.swift
```

```
view.backgroundColor = UIColor(named: .Background)
```

```
button.tintColor = UIColor(named: .Movie)
```

...

Second

TEX

STYLES

THE FIVE BOXING WIZARDS JUMP QUICKLY.

THE FIVE BOXING WIZARDS JUMP QUICKLY.

THE FIVE BOXING WIZARDS JUMP QUICKLY.

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean.

A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradigmatic country, in which roasted parts of sentences fly into your mouth.

Even the all-powerful Pointing has no control about the blind texts — it is an almost unorthographic life. One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar.

The Big Oxmox advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didn't listen. She packed her seven versalia, put her initial into the belt and made herself on the way.

DON'T

```
cell.textLabel.font = UIFont.systemFontOfSize(15)

UINavigationBar.appearance()
    .titleTextAttributes = [
        NSFontAttributeName: UIFont.systemFontOfSize(15),
        ...
    ]
```

MODELING TEXT STYLES

- ▶ Font Name, Size, Weight
- ▶ Line Height, Kerning, Shadow
- ▶ Dynamic Type

MODELING TEXT STYLES

```
enum TextStyle {  
    case dynamic(style: UIFontTextStyle)  
  
    case custom(  
        size: CGFloat,  
        weight: CGFloat,  
        height: CGFloat?,  
        dropShadow: Bool  
    )  
}
```

ADDING PREDEFINED STYLES

```
extension TextStyle {  
  
    static let body = TextStyle.dynamic(style: .body)  
  
    static let headline = TextStyle.custom(size: 20,  
                                           weight: UIFontWeightSemibold,  
                                           height: 24,  
                                           dropShadow: true)  
  
    ...  
}
```

TEXT STYLE ADDITIONS

```
extension TextStyle {  
  
    var font: UIFont {  
        switch self {  
            case let .dynamic(style):  
                return UIFont.preferredFont(forTextStyle: style)  
            case let .custom(size, weight, _, _):  
                return UIFont.systemFont(ofSize: size, weight: weight)  
        }  
    }  
  
    var stringAttributes: [String : AnyObject] {  
        ...  
    }  
}
```

USING TEXT STYLES

```
UINavigationBar.appearance()  
    .titleTextAttributes = TextStyle.headline.stringAttributes  
  
button.titleLabel?.font = TextStyle.callout.font  
  
let headlineLabel = Label(textStyle: .callout)  
let titleLabel = Label(textStyle: .body)  
let metadataLabel = Label(textStyle: .caption)
```

Third

CUSTOMIZING CONTROLS

ACTION

ACTION

ACTION

ACTION

GET TO KNOW YOUR TOOLS

- ▶ Image slicing
- ▶ Template rendering mode

Common > Resources > Assets.xcassets > ButtonBackgroundOutline

Image Set
Name: ButtonBackgroundOutline

Devices
All Universal
iOS iPhone
 iPad
OS X Mac
TV OS Apple TV
watchOS Apple Watch

Width: Any

Height: Any

Memory: 1 GB
 2 GB
 4 GB

Graphics: Metal 1v2
 Metal 2v2
 Metal 3v1

Scale Factors: Multiple

Render As: **Template Image**

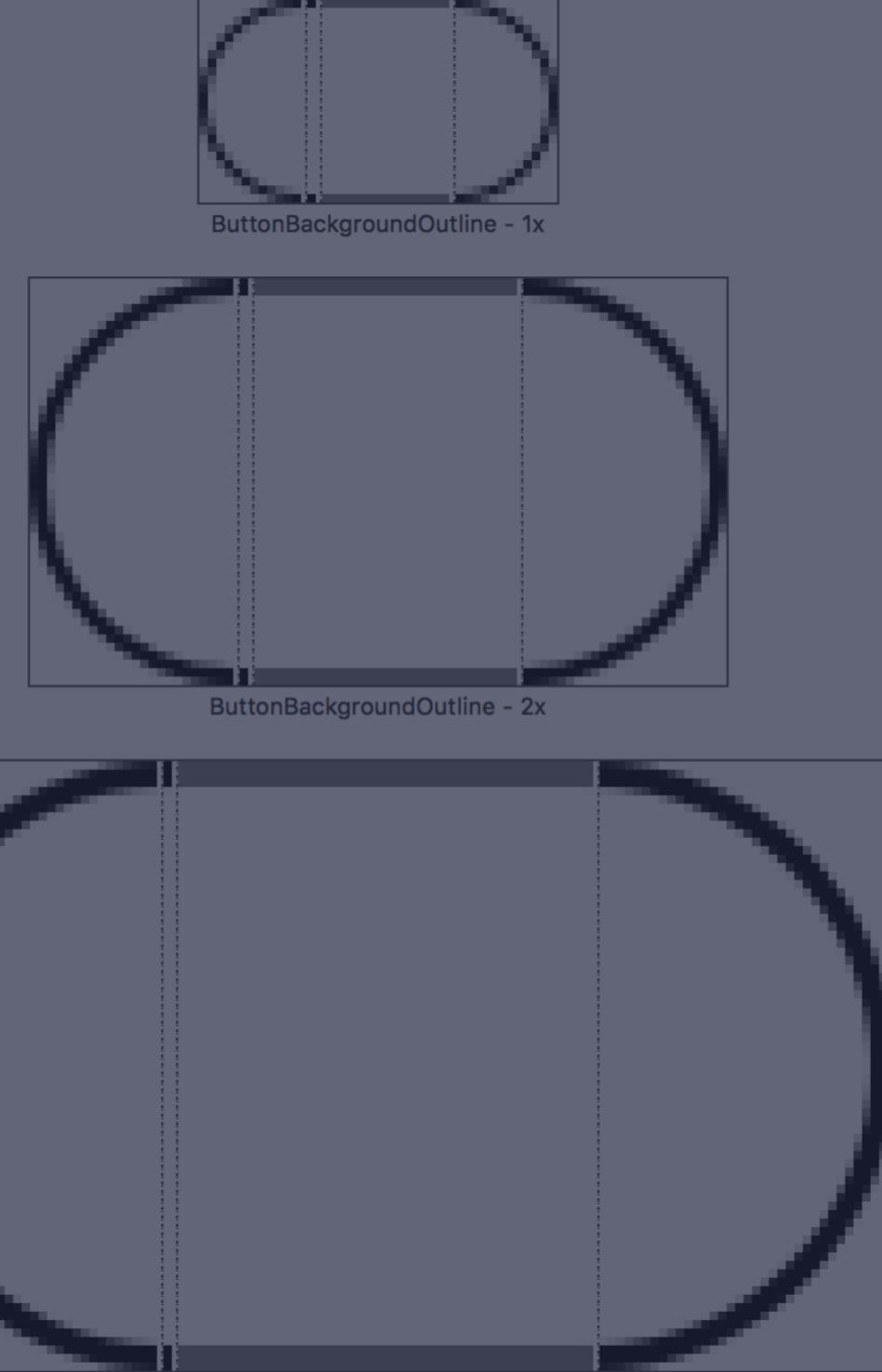
On Demand Resource Tags: Tags

Filter: Show Overview

ButtonBackgroundOutline - 1x

ButtonBackgroundOutline - 2x

ButtonBackgroundOutline - 3x



MODELING BUTTON STYLES

```
final class Button: UIButton {  
  
    struct Style {  
        let titleColor: UIColor?  
        let normalBackgroundImage: UIImage?  
        let highlightedBackgroundImage: UIImage?  
    }  
}
```

ADDING PREDEFINED BUTTON STYLES

```
extension Button.Style {  
  
    static let outline = Button.Style(  
        titleColor: nil,  
        normalBackgroundImage: UIImage(asset: .ButtonBackgroundOutline),  
        highlightedBackgroundImage: UIImage(asset: .ButtonBackgroundHighlighted),  
    )  
  
    static let solid = Button.Style(  
        titleColor: UIColor(named: .Text1),  
        normalBackgroundImage: UIImage(asset: .ButtonBackgroundSolid),  
        highlightedBackgroundImage: nil  
    )  
}
```

USING OUR CUSTOM BUTTONS

```
let button = Button(title: "ADD TO...",  
                    style: .outline,  
                    icon: .add)
```

```
button.style = .solid  
button.icon = .checkmark  
button.title = "WATCHED"  
button.tintColor = UIColor(named: .show)
```

+ ADD TO...

✓ WATCHED

Fourth

DISPLAYING IMAGES

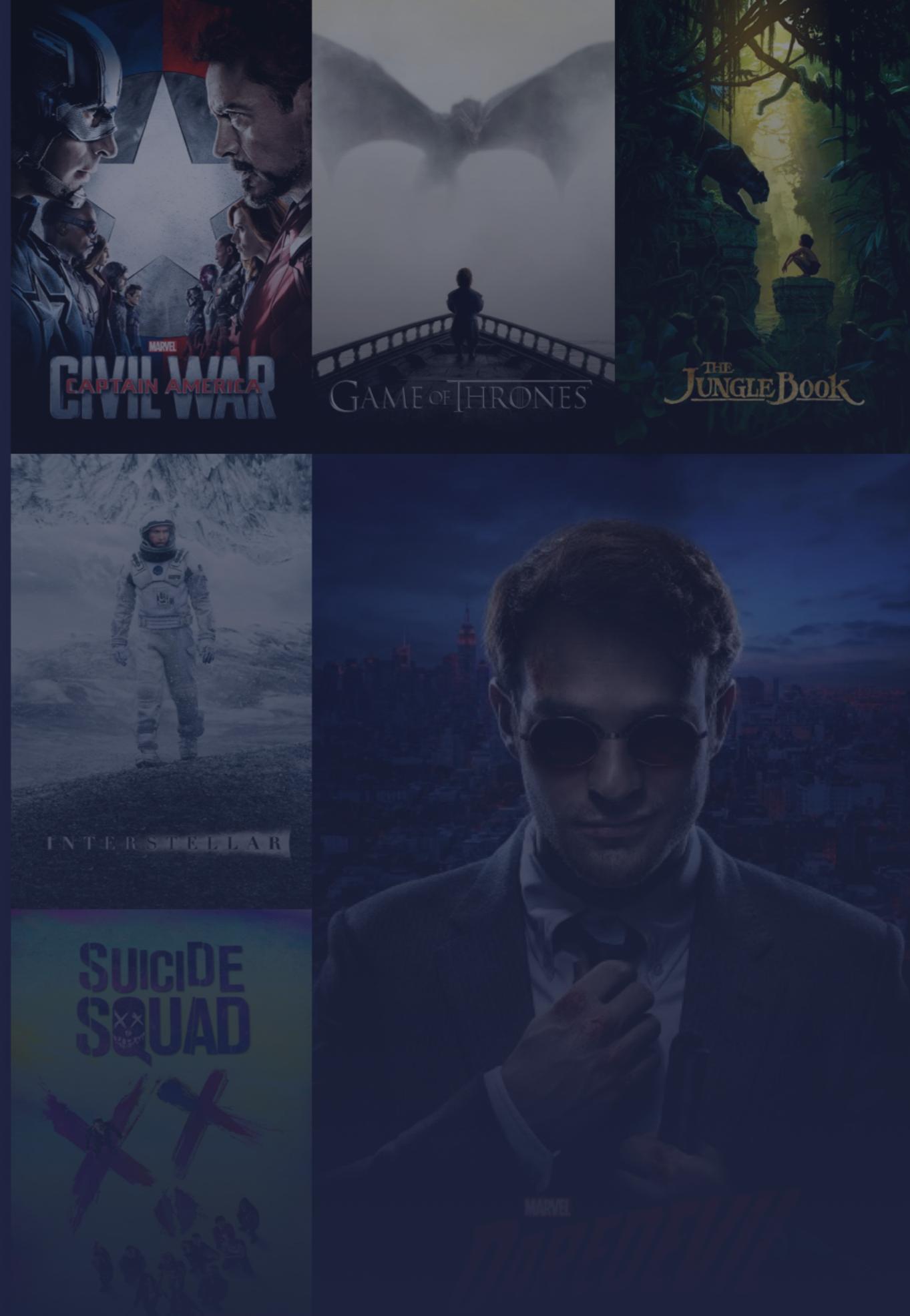


IMAGE REQUIREMENTS

- ▶ Download and decode off the main thread
 - ▶ Post-process off the main thread
- ▶ Cache original and post-processed images



PINRemoteImage

github.com/pinterest/PINRemoteImage

BACKDROPS



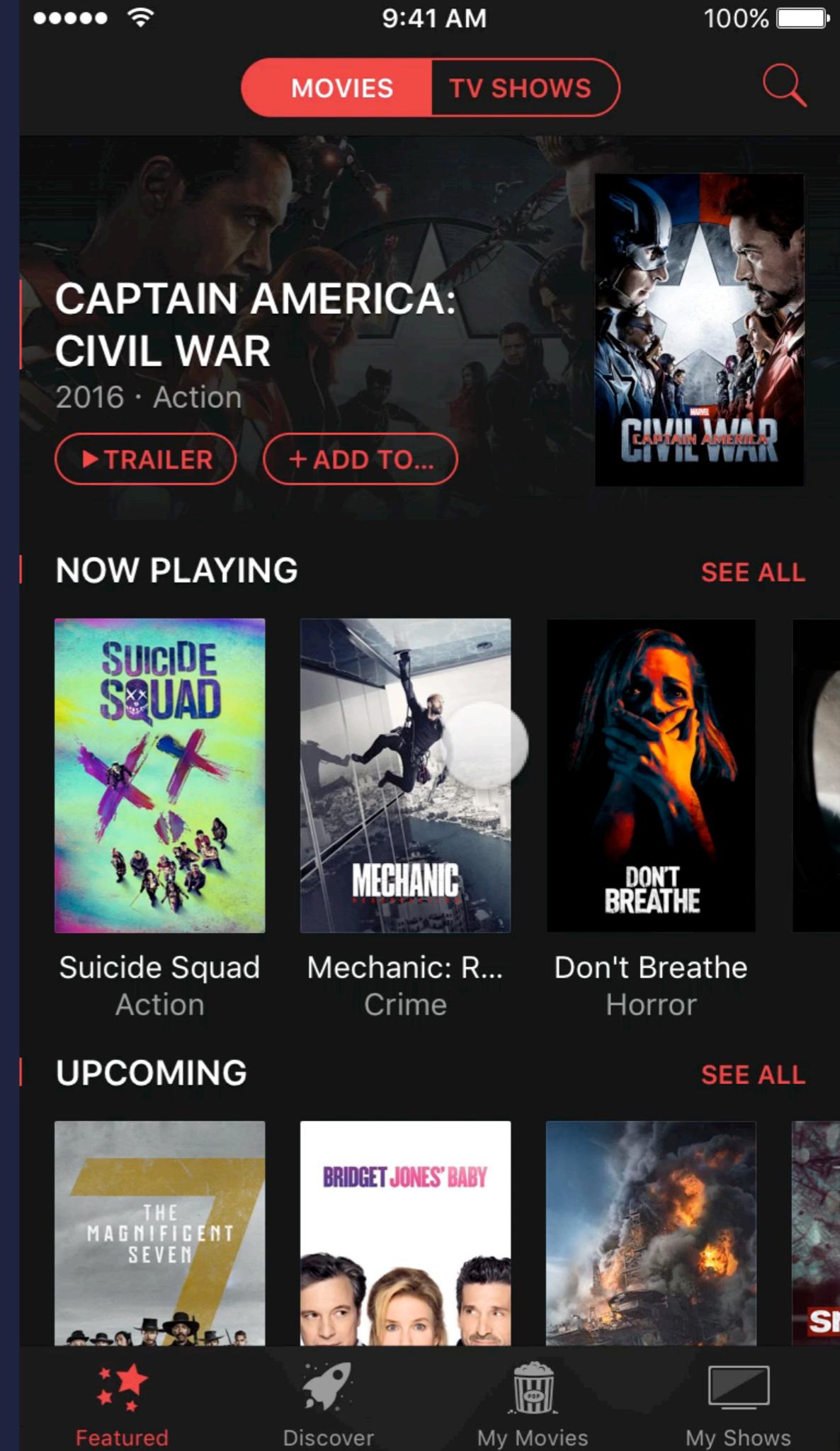
BACKDROPS

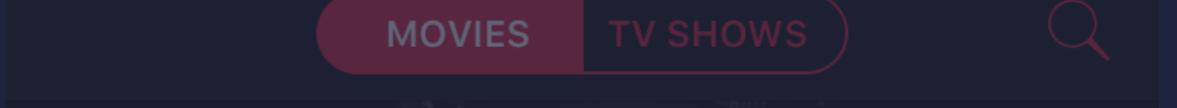
```
imageView.pin_setImageFromURL(  
    imageURL,  
    processorKey: "backdrop"  
) { result, _ in  
    return result.image?.composited(  
        withImages: [  
            (overlayGradient, .Overlay, 1),  
            (gradient, .Normal, 1)  
        ]  
    )  
}
```



REUSING VIEW CONTROLLERS

Fifth





**CAPTAIN AMERICA:
CIVIL WAR**
2016 · Action
▶ TRAILER + ADD TO...

ItemStripViewController<T>

```
init(  
    resource: Resource,  
    configure: (ItemCell, T) -> ()  
)
```

NOW PLAYING SEE ALL

SUICIDE SQUAD
Mechanic
DON'T BREATHE

Suicide Squad Mechanic: R... Don't Breathe
Action Crime Horror

UPCOMING SEE ALL

THE MAGNIFICENT SEVEN
BRIDGET JONES' BABY
SN

Featured Discover My Movies My Shows

MOVIES TV SHOWS

CAPTAIN AMERICA: CIVIL WAR
2016 · Action

► TRAILER + ADD TO...

NOW PLAYING SEE ALL

SUICIDE SQUAD

Mechanic: Resurrection

DON'T BREATHE

UPCOMING SEE ALL

THE MAGNIFICENT SEVEN

BRIDGET JONES' BABY

SN

Featured Discover My Movies My Shows

The image shows a mobile application interface for a movie database. At the top, there are tabs for 'MOVIES' and 'TV SHOWS', with 'MOVIES' being the active tab. A search icon is also at the top right. The main content area features a large card for 'CAPTAIN AMERICA: CIVIL WAR' (2016, Action), which includes a trailer button and an 'ADD TO...' button. Below this, there are two sections: 'NOW PLAYING' and 'UPCOMING'. The 'NOW PLAYING' section displays three movies: 'SUICIDE SQUAD', 'Mechanic: Resurrection', and 'DON'T BREATHE'. The 'UPCOMING' section displays three movies: 'THE MAGNIFICENT SEVEN', 'BRIDGET JONES' BABY', and another partially visible movie. At the bottom, there are navigation icons for 'Featured', 'Discover', 'My Movies', and 'My Shows'.

ItemCarouselViewController<T>

```
init(  
    resource: Resource,  
    configure: (CardItemCell, T) -> ()  
)
```

MOVIES

TV SHOWS



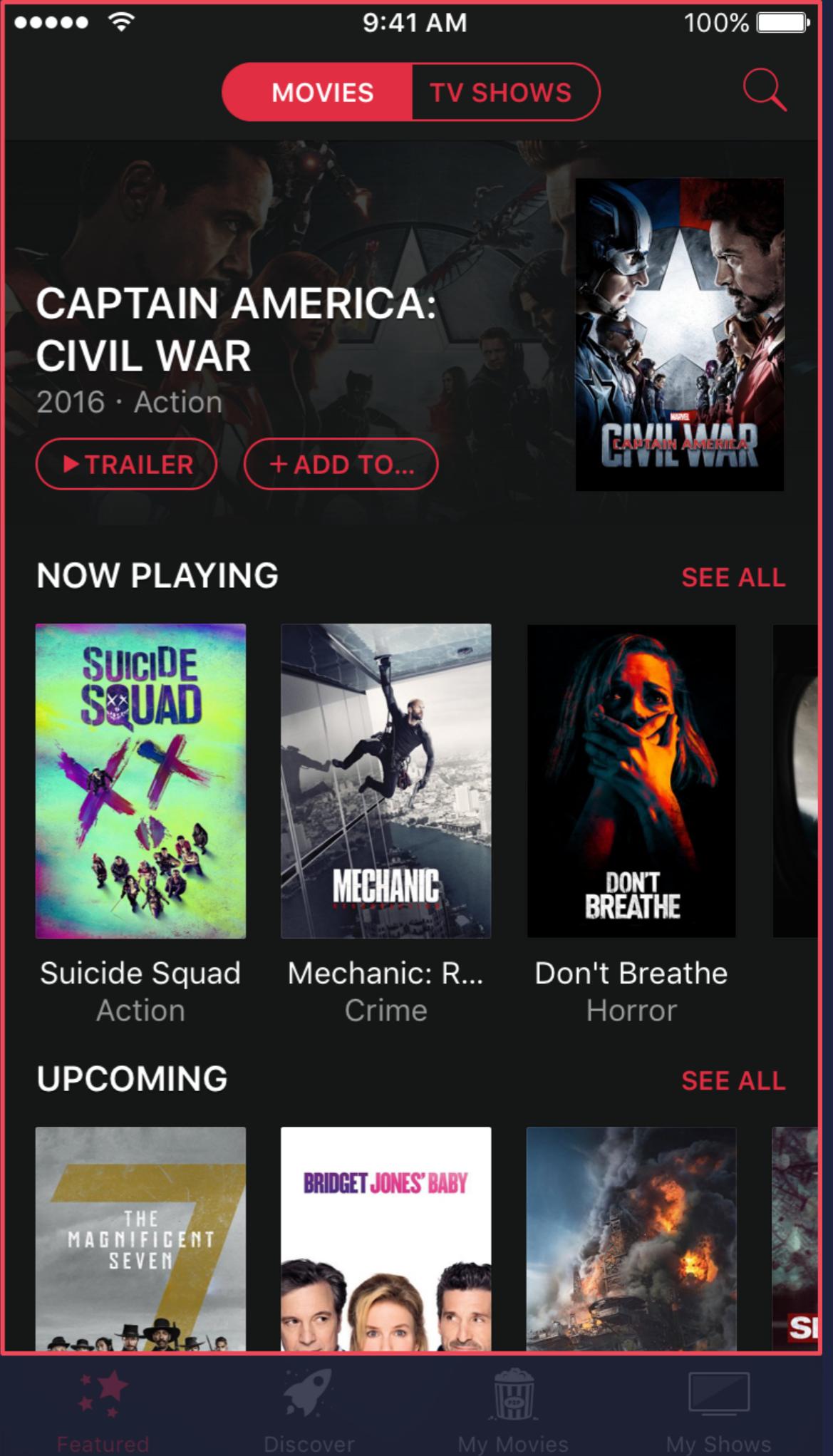
StackViewController

```
init(  
    viewControllers: [UIViewController])
```

The image shows a mobile application interface for a movie service. At the top, there are tabs for 'MOVIES' (selected) and 'TV SHOWS'. A search bar is located at the top right. Below the tabs, a large movie card for 'CAPTAIN AMERICA: CIVIL WAR' (2016 · Action) is displayed, featuring a collage of characters and a 'TRAILER' and '+ ADD TO...' button. To the right of the main card is a smaller image of the movie poster. Below this section are two buttons: 'NOW PLAYING' and 'SEE ALL'. Under 'NOW PLAYING', three movie cards are shown: 'SUICIDE SQUAD' (Action), 'MECHANIC' (Crime), and 'DON'T BREATHE' (Horror). Each card includes the movie title, genre, and a small image. Below this is another section titled 'UPCOMING' with 'SEE ALL' to its right. Three movie cards are visible: 'THE MAGNIFICENT SEVEN' (Action), 'BRIDGET JONES' BABY' (Romantic), and 'SI' (Science Fiction). At the bottom, there are four navigation icons: 'Featured' (with a star icon), 'Discover' (with a rocket icon), 'My Movies' (with a popcorn icon), and 'My Shows' (with a television icon).

SegmentedViewController

```
init(  
    viewControllers: [UIViewController]  
)
```



REUSING VIEW CONTROLLERS

- ▶ Forget Storyboards
- ▶ Separate navigation logic
- ▶ Separate business logic
- ▶ Configurable

A dark, moody photograph of a man with light hair, wearing a red tank top and a red headband. He is looking down with a somber expression, his hand resting near his face. The background is a textured, dark surface.

“TANK” YOU!!

@PUMPKIN + @GONZALEZREAL

ACKNOWLEDGEMENTS

- ▶ iOS App icon template by Sketch Team
- ▶ 12 Free Reaction Emojis by Marc Gonzales
- ▶ Making sense of MVP by Henrik Kniberg
 - ▶ iOS Devices by Robbie Pearce
- ▶ “Tank” You from Kung Fury by David Sandberg

MADE IN @DECKSETAPP + @SKETCHAPP