

< / >

Swift IAP

Day 01

Designed by
CARYL , SEAN , GLEN

Date
14 JAN 2021

What is Swift?



- Programming language created by Apple in June 2014
- Used for building applications on macOS, iOS, watchOS, tvOS and beyond.
- Open source

For more information, visit <https://developer.apple.com/swift/>

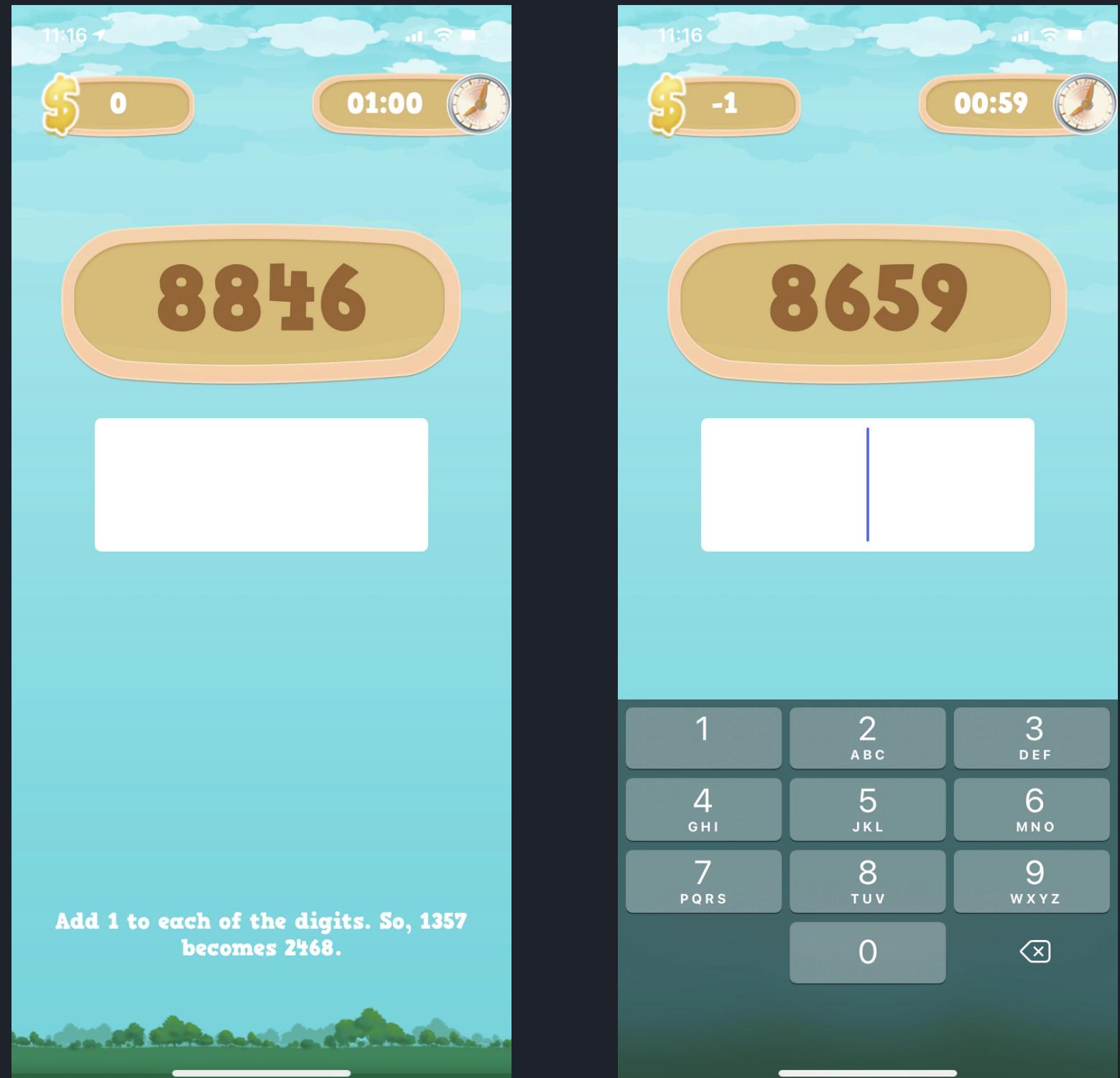
Goal for today's session:

Add 1

WHAT IS ADD 1?

The goal of the game is to *Add 1* to each of the digits of a randomly generated 4-digit number as many times as possible in 60 seconds.

Fun fact: This game was inspired by the Add 1 mental exercise described in the book *Thinking, Fast and Slow* by Daniel Kahneman to test for cognitive strain.



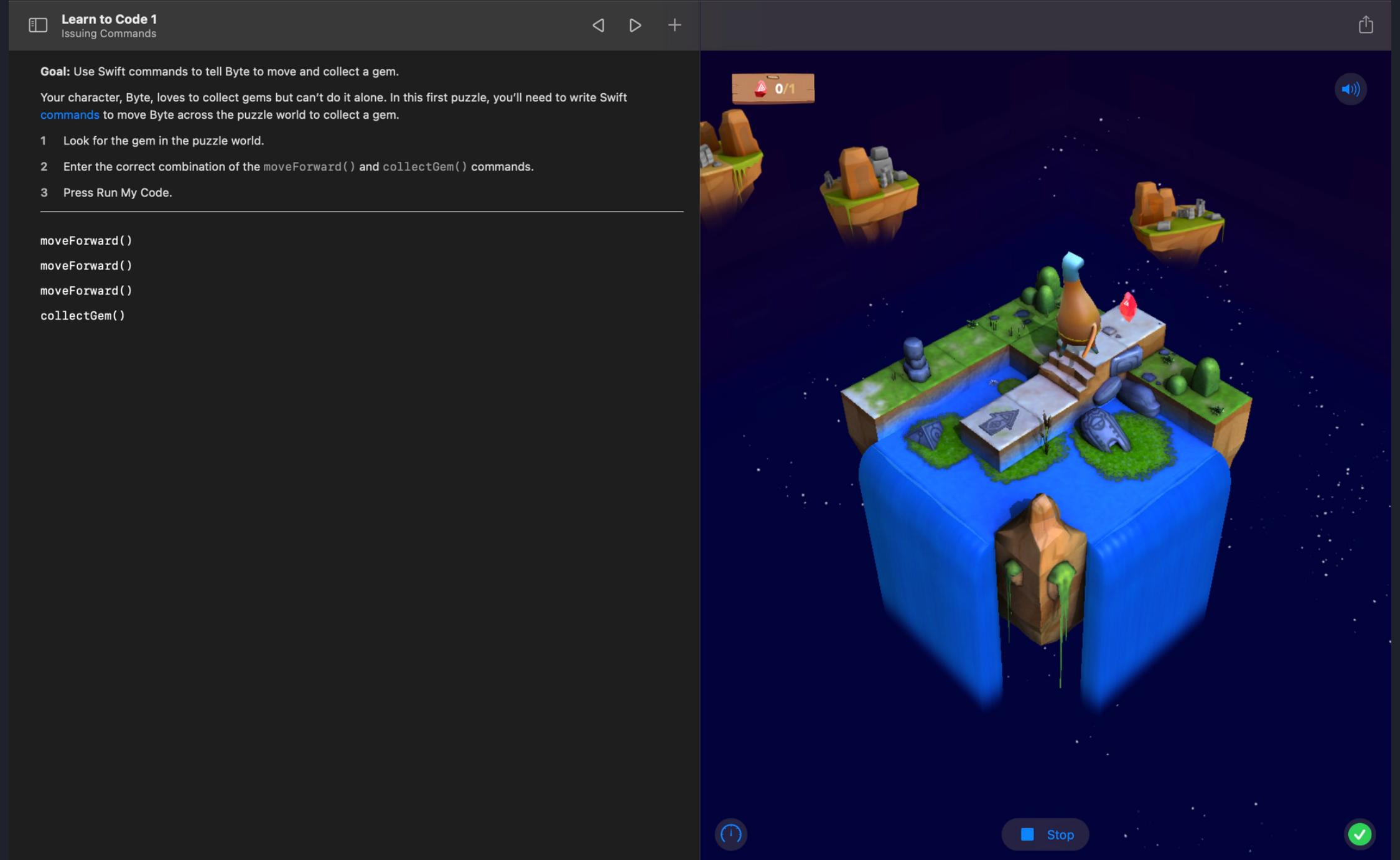
Agenda

- Programming Basics
- Fiddling with Xcode
- Code-Along UI
- Code-Along everything else

< / >

Before we get started with Xcode...

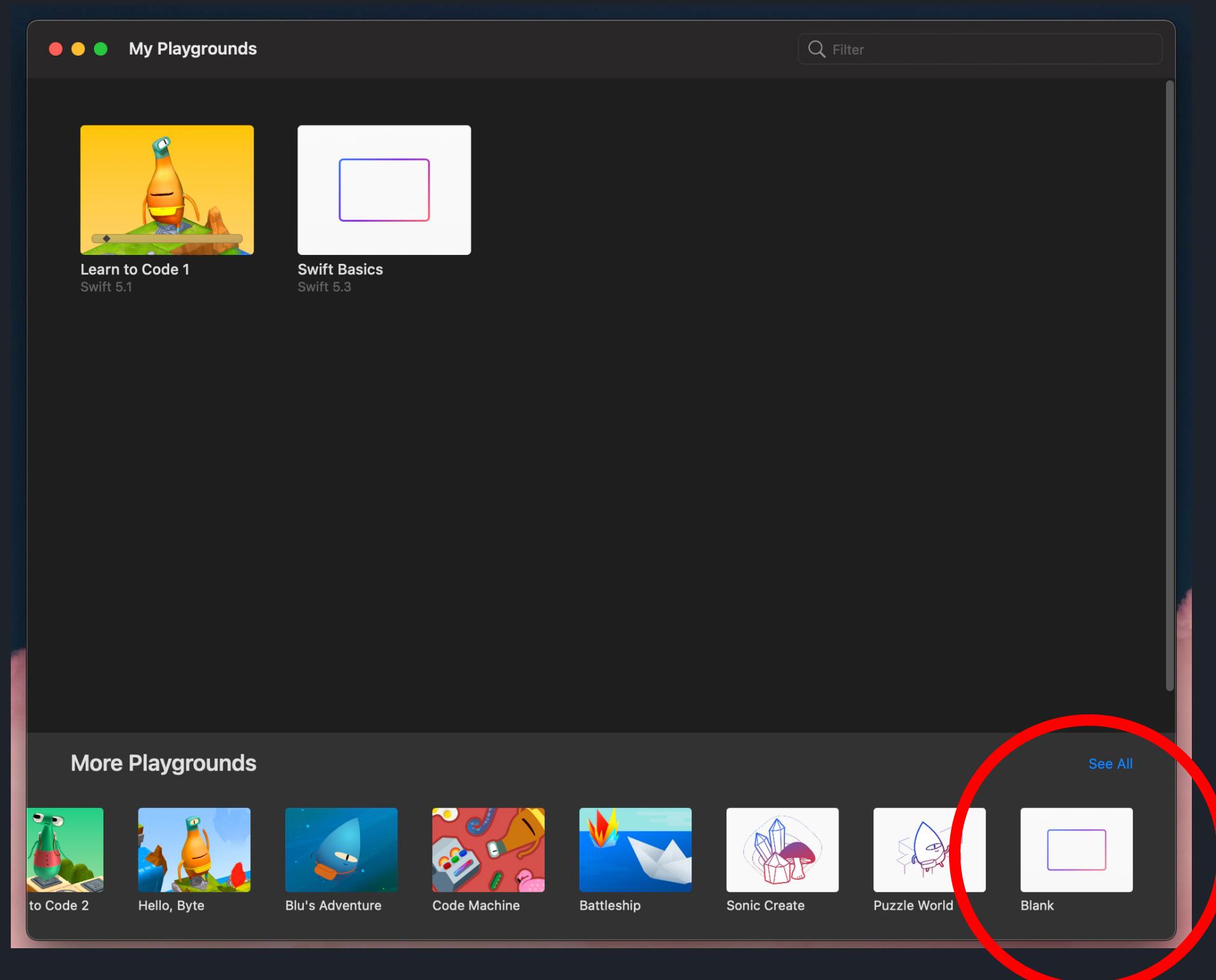
Swift Playgrounds



- Swift Playgrounds is an educational tool and development environment for Swift.
- Learn programming through solving puzzles - ideal for those starting to learn programming
- Available on both iPad and Mac

For more information: <https://www.apple.com/swift/playgrounds/>

Create new playground



Select **Blank** playground

< / >

Swift Programming Basics

Declaring Constants & Variables

Constants and variables must be declared before they're used. You declare **constants** with the **let** keyword and **variables** with the **var** keyword.

```
let maximumNumberOfLoginAttempts = 10 // Constant value → Cannot be modified!  
var currentLoginAttempt = 0 // Can be modified!
```

Type Annotations

You can provide a *type annotation* when you declare a constant or variable, to be clear about the kind of values the constant or variable can store.

```
var petOne: String  
var petTwo: String = "Fluffy"
```

Type Inference

Swift is **type-safe**. A type safe language encourages you to be clear about the types of values your code can work with. This means that Swift performs **type checking** at **compile time**.

If you don't specify the type of value you need, Swift uses **type inference** to work out the appropriate type. Type inference enables a compiler to deduce the type of a particular expression automatically when it compiles your code, simply by examining the values you provide.

WILL COMPILE AND RUN

```
let numeroUno: Int = 4 // This is valid
let numeroDos = 6 // This is valid too!
let sum = numeroUno + numeroDos
print(sum) // Output: 10
```

RESULTS IN COMPILE TIME ERROR

```
let numeroUno: Int = 4
let numeroDos = "6"
let sum = numeroUno + numeroDos // ERROR HERE
print(sum)
```

String Interpolation

```
// String interpolation in Swift
let name = "Steve"
let message = "Hello \(name)"
print(message) // Output: Hello Steve
```

For Loops

```
// For loop in Swift
for index in 1...5 {
    print("Iteration number: \(index)")
```

What about while loops?

String Concatenation

```
var subString = "May Stack Overflow"
subString += " Be With You"
print(subString) // Output: May Stack Overflow Be With You
```

Functions in Swift

The following is the structure of a function declaration in Swift.

```
func functionName(parameter name: parameter type) → return type {  
    // Body of function  
}
```

Optionals

Optionals are used in situations where a value may be absent (i.e. nil). An optional represents two possibilities: Either there *is* a value, and you can **unwrap** the optional to access that value, or there *isn't* a value at all and will be set to nil.

```
var pillarName: String? // Set to nil by default
print(pillarName) // Output: nil
pillarName = "ISTD"
print(pillarName!) // Must unwrap optional with ! before using
```

Unwrapping Optionals

The purpose of optionals is to ensure that your application does not crash if a value is absent. Hence, it is important to unwrap it properly.

```
var pillarName: String? // Set to nil by default
print(pillarName) // Output: nil
pillarName = "ISTD"
print(pillarName!) // Must unwrap optional with ! before using
var pillarName: String?
if let safePillarName = pillarName { // Optional Binding
    print(safePillarName)
}
guard let safePillarName = pillarName else {return} // Guard Let; Similar to Optional Binding but
allows you to throw an error
print(pillarName ?? "Pillar Name is Nil") // Nil Coalescing Operator
print(pillarName?.count) // Optional Chaining
pillarName = "ISTD"
print(pillarName!) // Force Unwrapping
```

Let's get started with Xcode!





Choose a template for your new project:

Multiplatform

iOS

macOS

watchOS

tvOS

Other

Filter

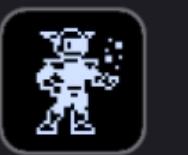
Application



App



Document App



Game



Augmented
Reality App



Sticker Pack App



iMessage App

Framework & Library



Framework



Static Library



Metal Library

Cancel

Previous

Next

Choose options for your new project:

Product Name:

Team: Caryl Peneyra (Personal Team)

Organization Identifier: cpaneyra.personal

Bundle Identifier: cpaneyra.personal.Add-1

Interface:

Life Cycle:

Language:

Use Core Data

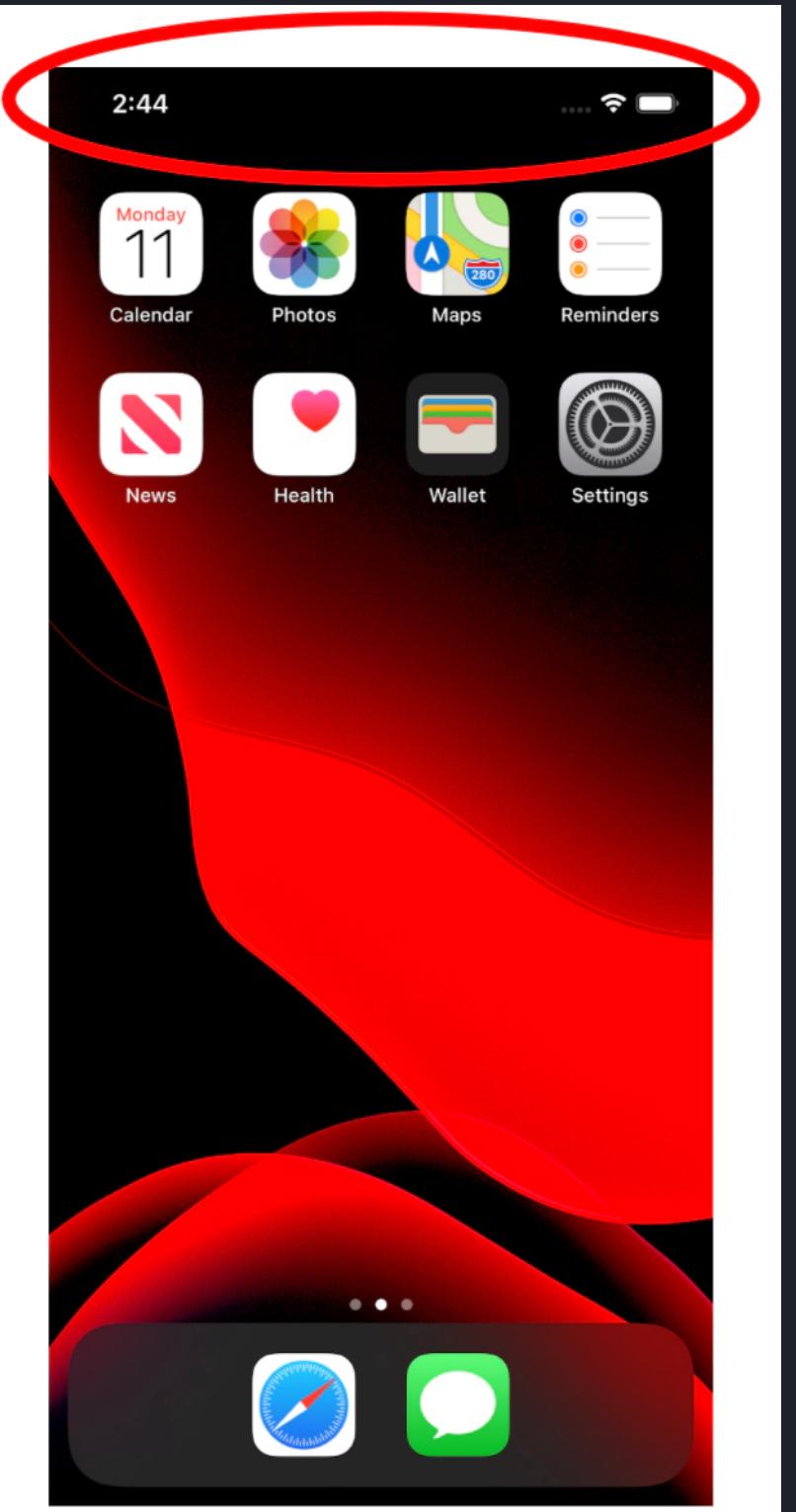
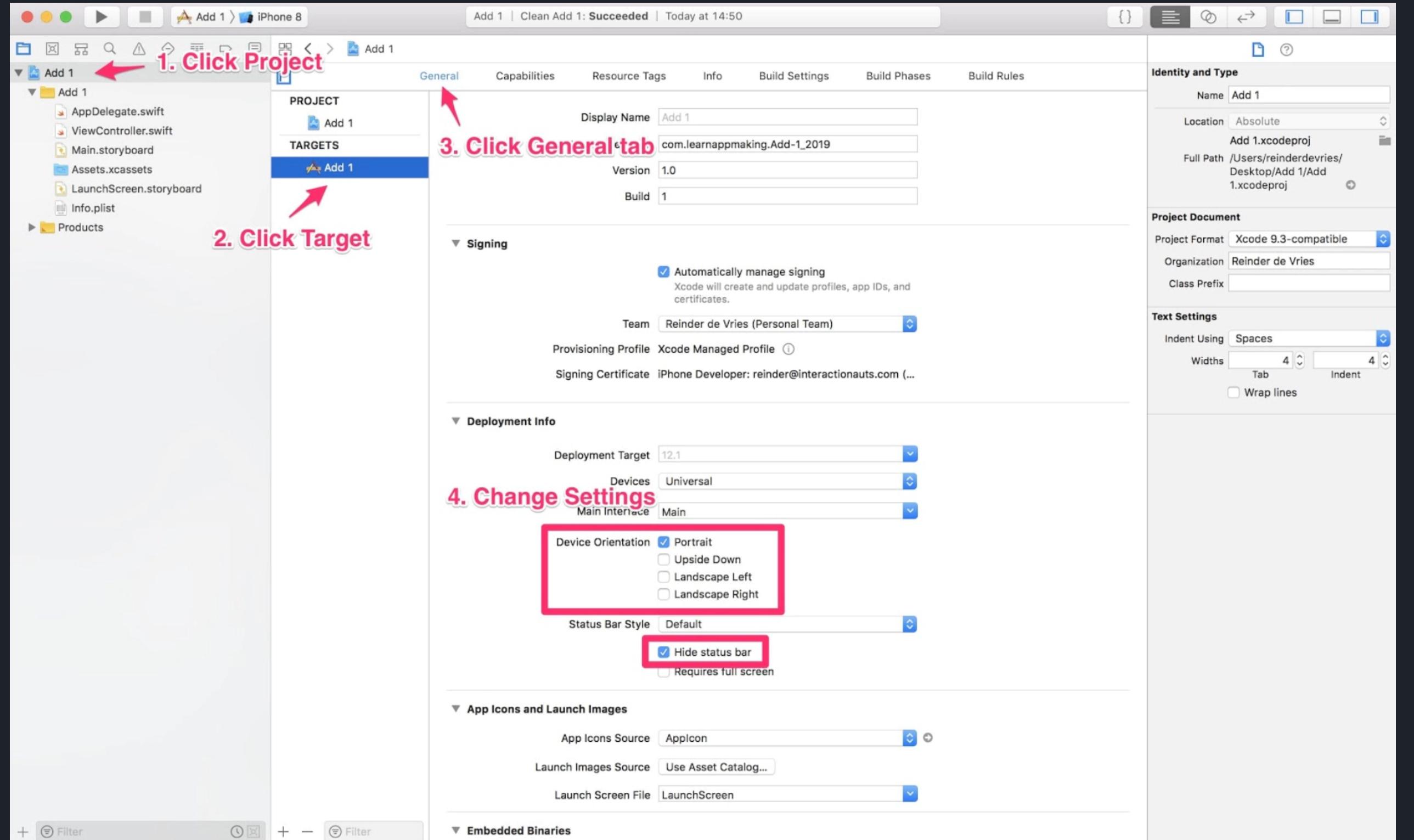
Host in CloudKit

Include Tests

Cancel

Previous

Next



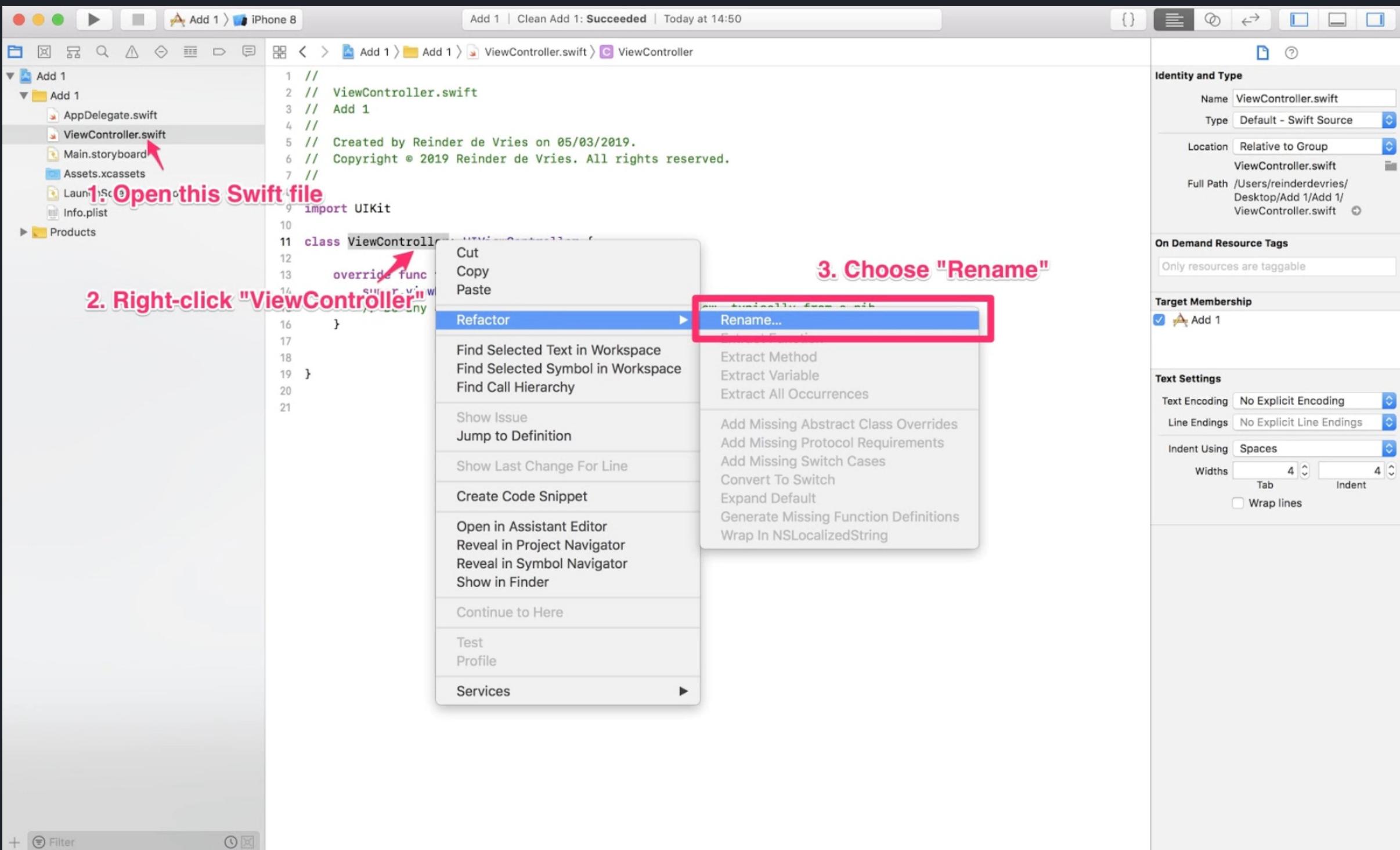
View Controller (VC)

WHAT IS A VIEW CONTROLLER?

- A View Controller manages a single User Interface (UI) / screen in your App
- Remember the Interface Builder (IB)? We can manage how the screen looks through IB or programmatically through the View Controller file

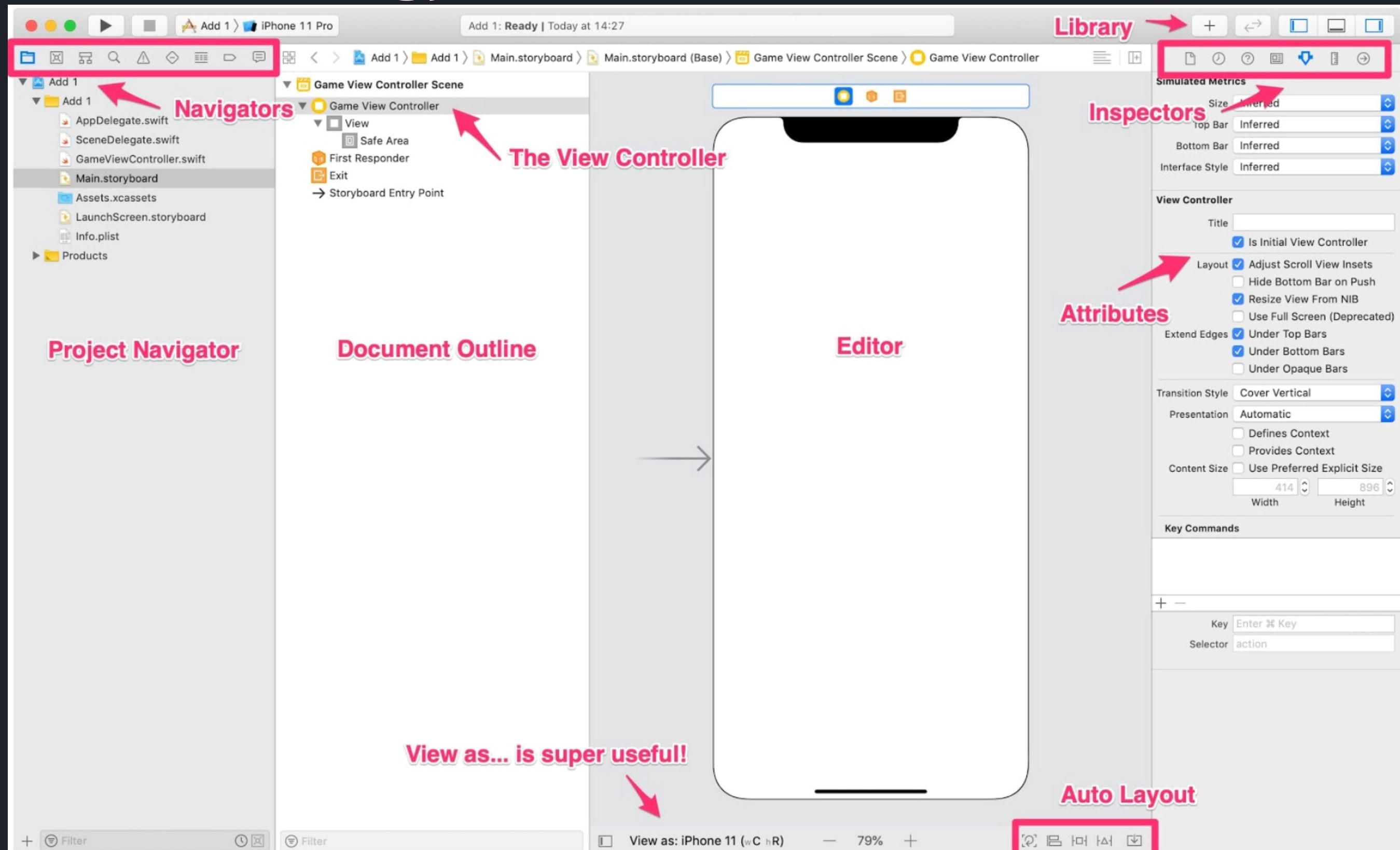
Refactoring

RENAMING VC

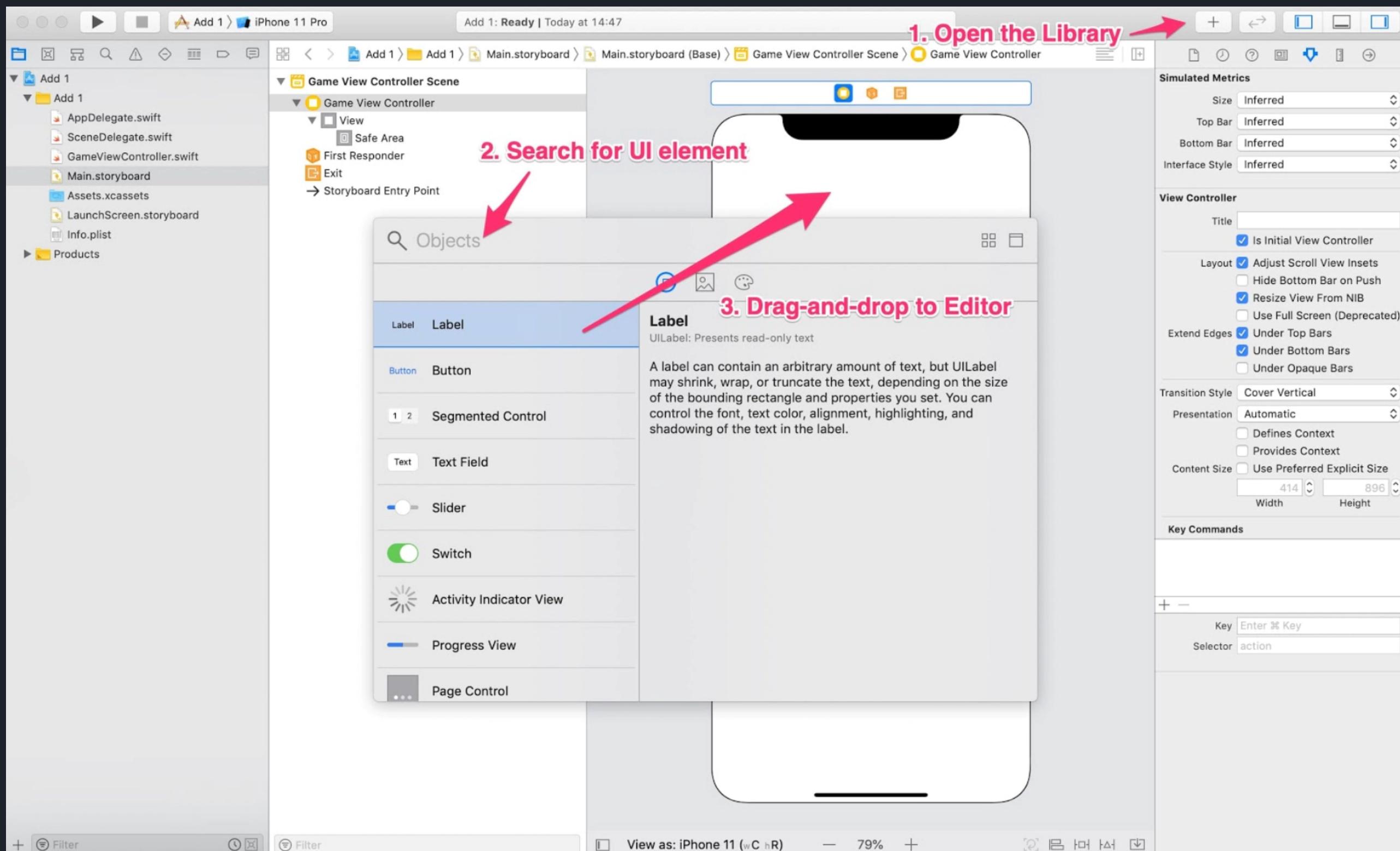


Yo lmao why don't
I just rename the
class directly

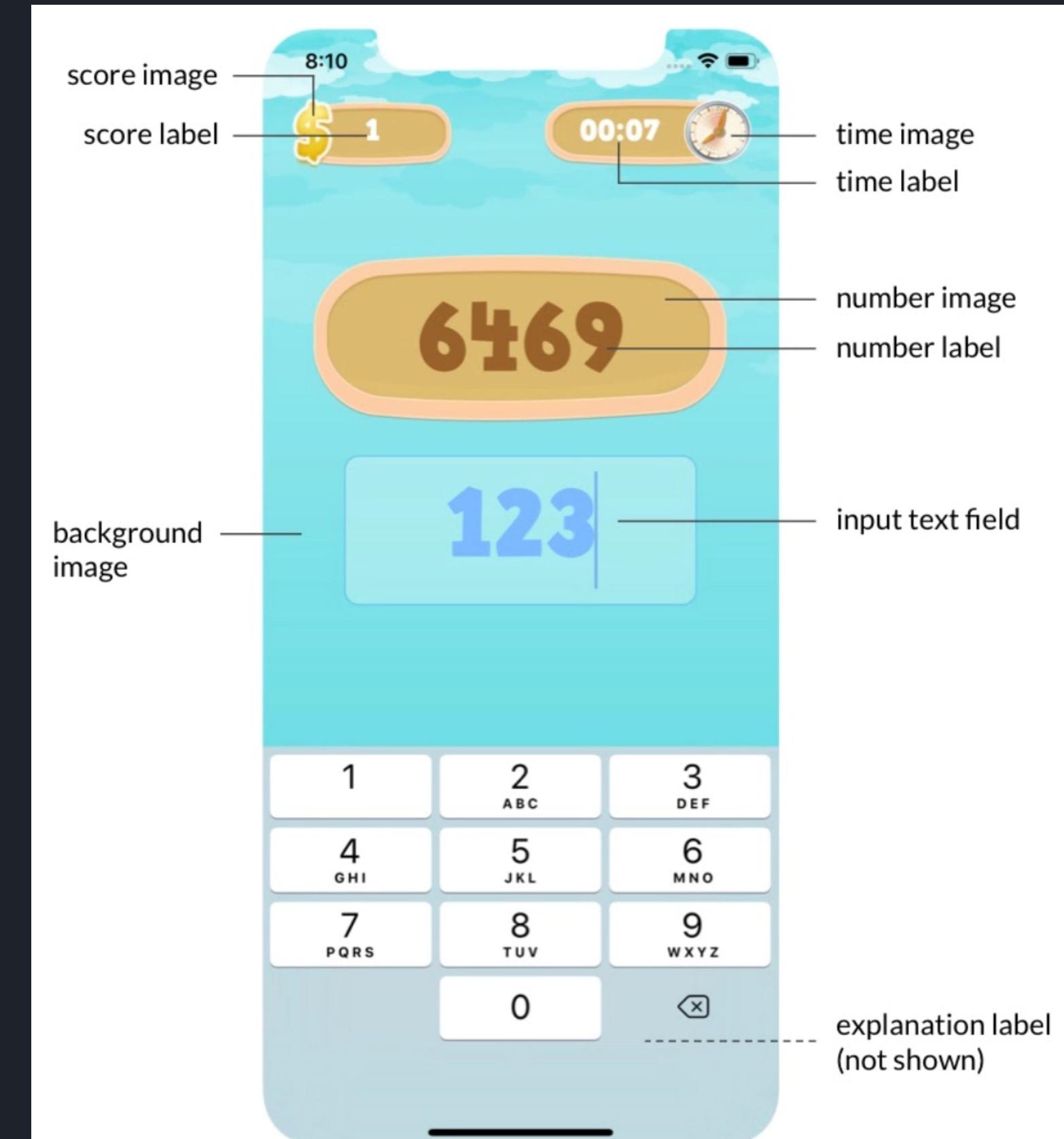
IB Terminology



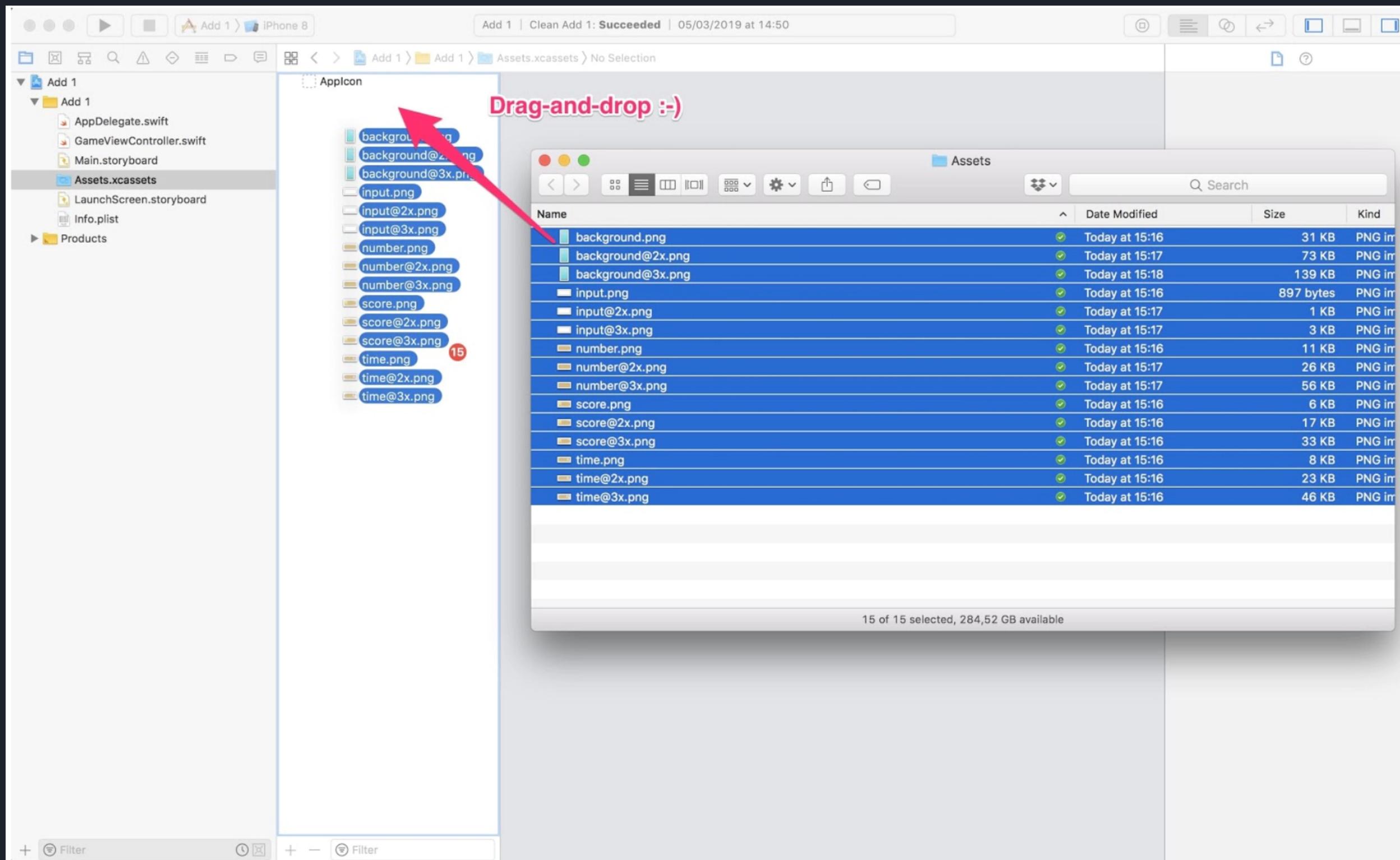
Adding Elements



UI End-Product



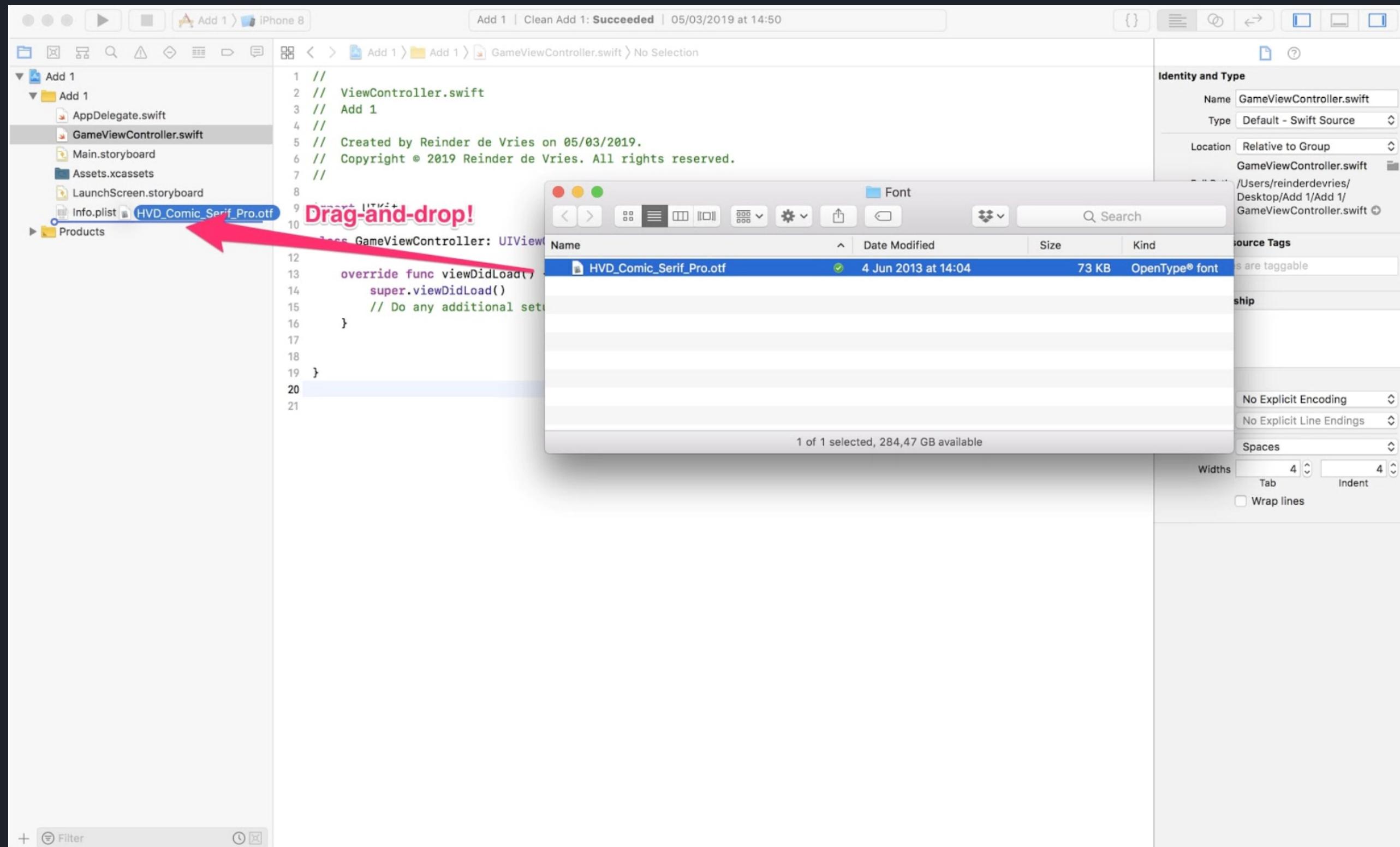
Adding Assets



What on earth is @2x and @3x ?! bruh

- Screen is made of pixels
- Different screen sizes have different pixels and **different pixel densities** (Pixels Per Inch, PPI)
- Screens with higher pixel densities require **higher resolution images**
- **Larger pictures = Higher resolution**
- Hence, @2x and @3x is an image at 2x and 3x the size respectively

Adding Fonts



Bruh please stop
what is this info.plist

Info.plist

PROPERTY LIST / PLIST

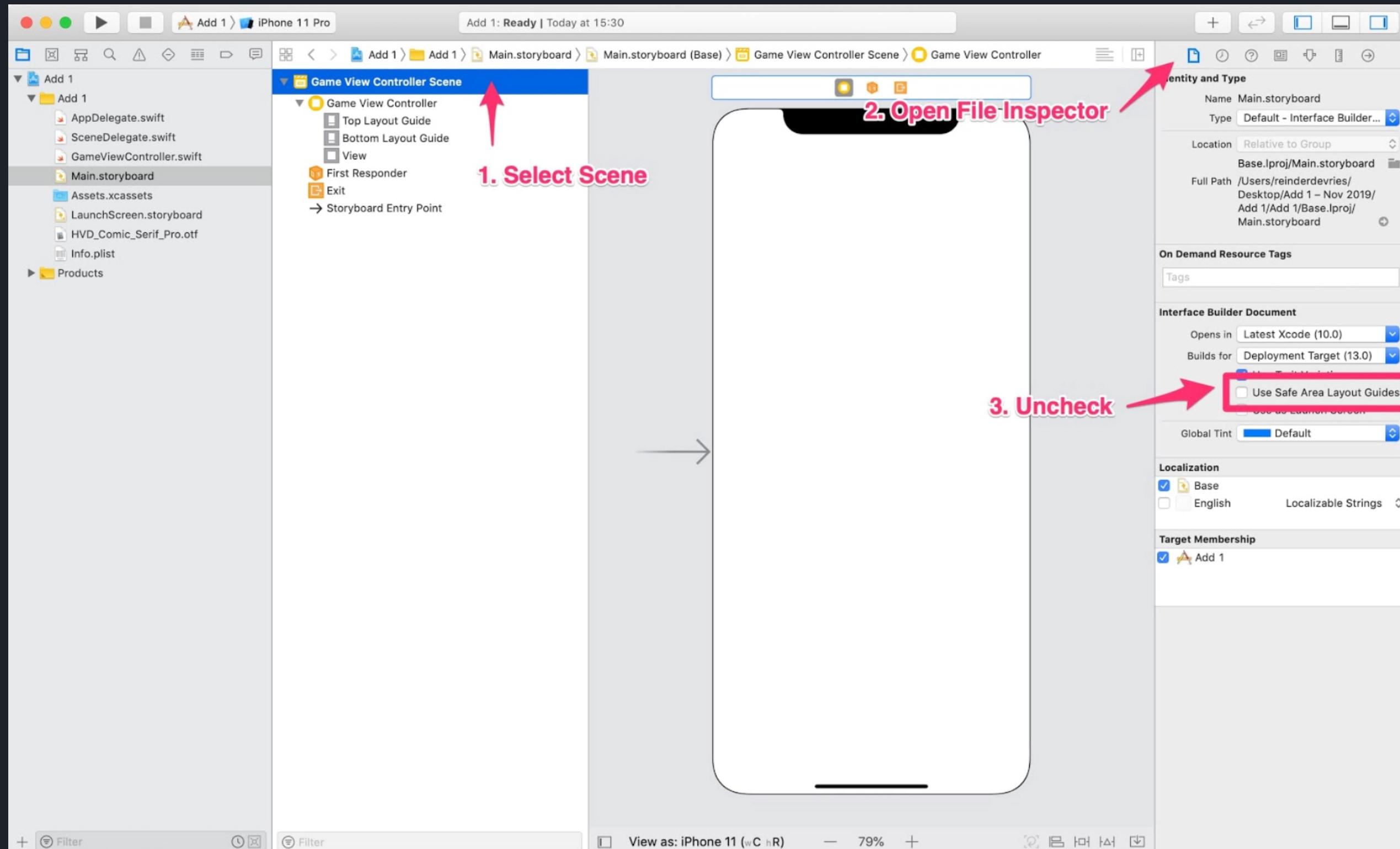
- A XML (Extensible Markup Language) file containing key-value data

WHY DO I NEED IT?

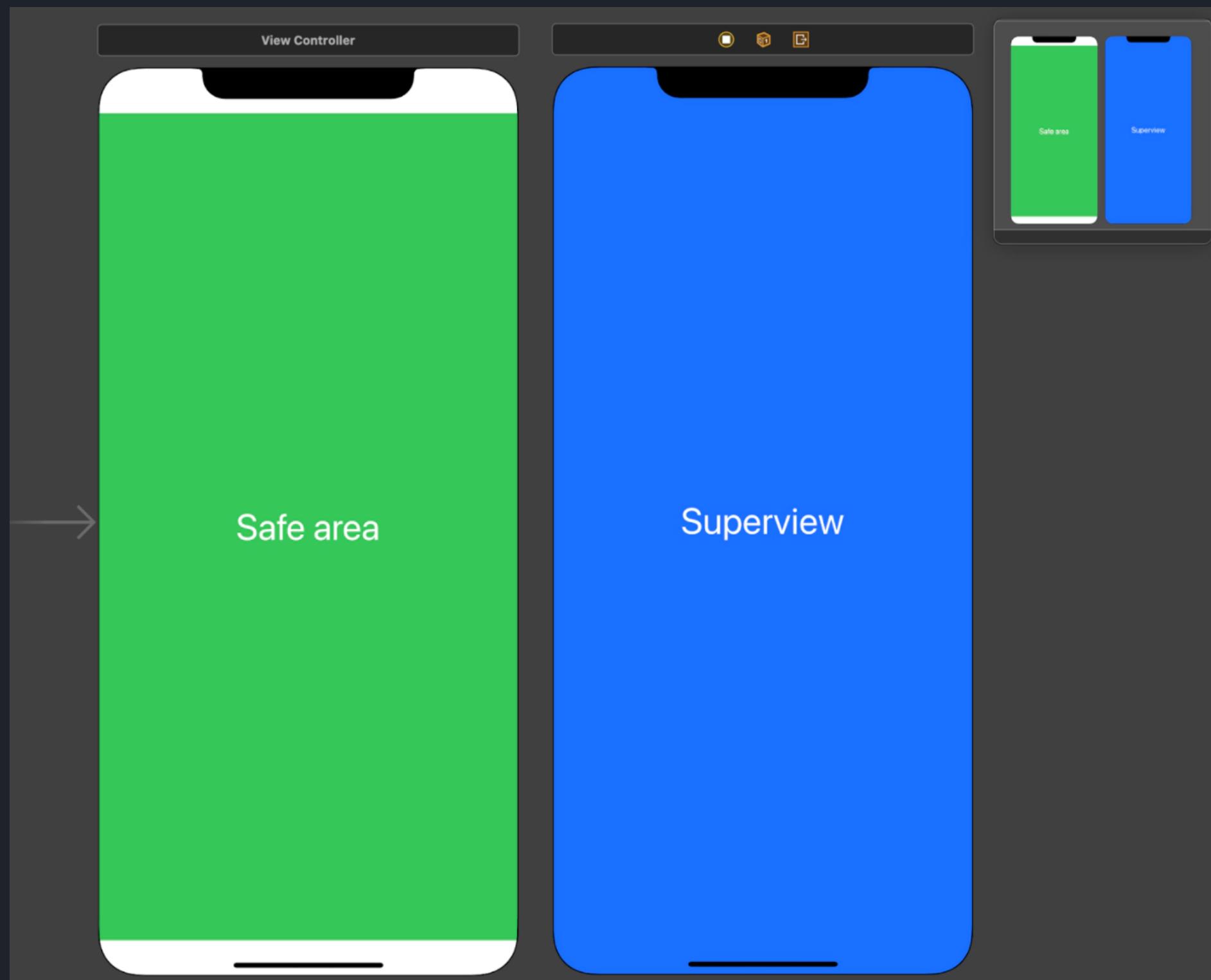
- Helps to store persistent data globally
 - Persistent = Even after application closes / restarts
 - Globally = Between different screens
- (For those familiar), it's like a .env file that accepts data types other than Strings

| Key | Type | Value |
|--|------------|---------------------------------|
| Localization native development region | String | \$(DEVELOPMENT_LANGUAGE) |
| Executable file | String | \$(EXECUTABLE_NAME) |
| Bundle identifier | String | \$(PRODUCT_BUNDLE_IDENTIFIER) |
| InfoDictionary version | String | 6.0 |
| Bundle name | String | \$(PRODUCT_NAME) |
| Bundle OS Type code | String | \$(PRODUCT_BUNDLE_PACKAGE_TYPE) |
| Bundle version string (short) | String | 1.0 |
| Bundle version | String | 1 |
| Application requires iPhone environment | Boolean | YES |
| Fonts provided by application | Array | (1 item) |
| Item 0 | String | HVD_Comic_Serif_Pro.otf |
| > Application Scene Manifest | Dictionary | (2 items) |
| Application supports indirect input events | Boolean | YES |
| Launch screen interface file base name | String | LaunchScreen |
| Main storyboard file base name | String | Main |
| > Required device capabilities | Array | (1 item) |
| Status bar is initially hidden | Boolean | YES |
| > Supported interface orientations | Array | (1 item) |
| > Supported interface orientations (iPad) | Array | (4 items) |

More IB Terminology



Safe Area VS Superview





Agenda Update!

Programming Basics

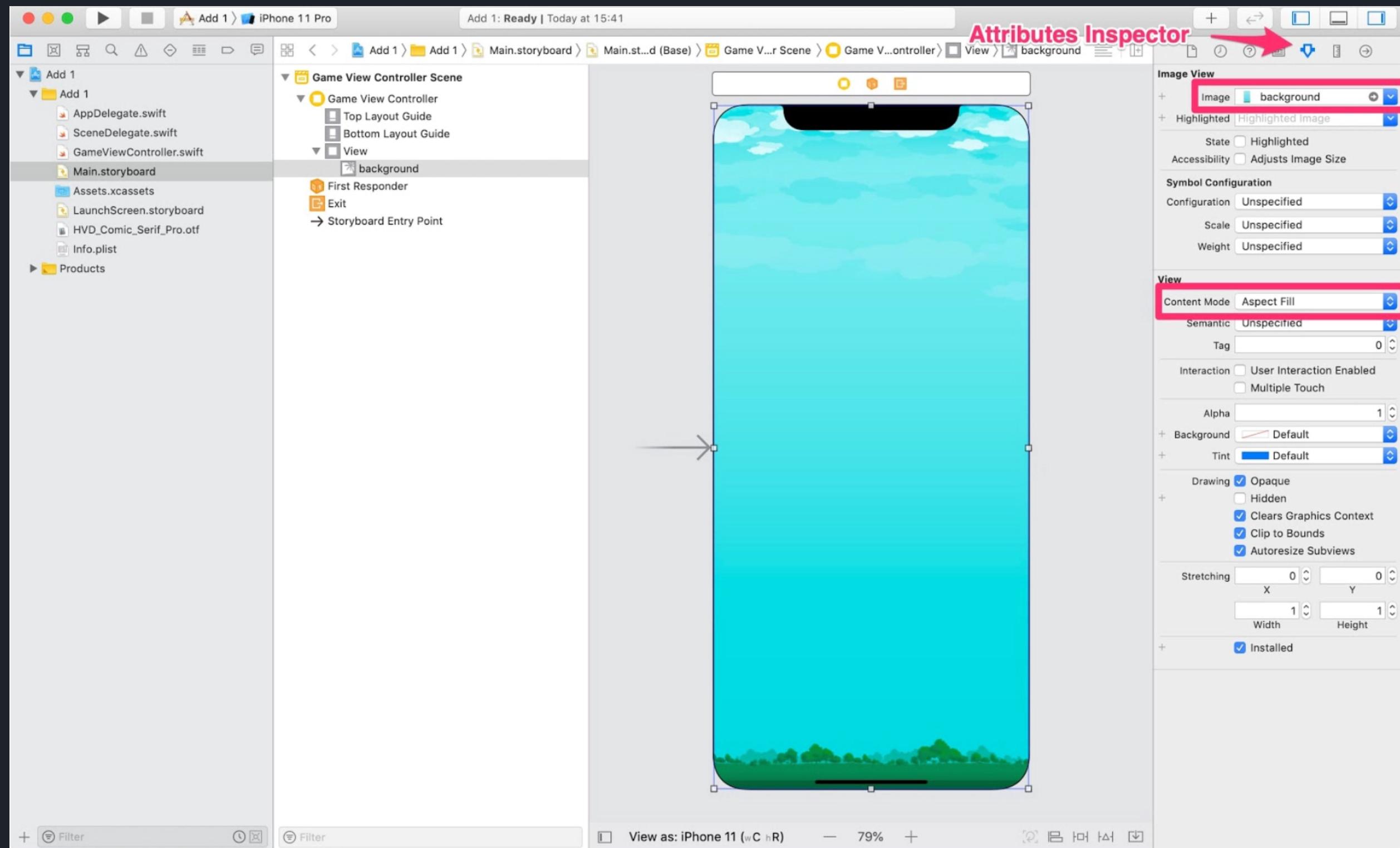
Python VS Swift!

Fiddling with Xcode

Code-Along UI

Code-Along everything else

Adding UI



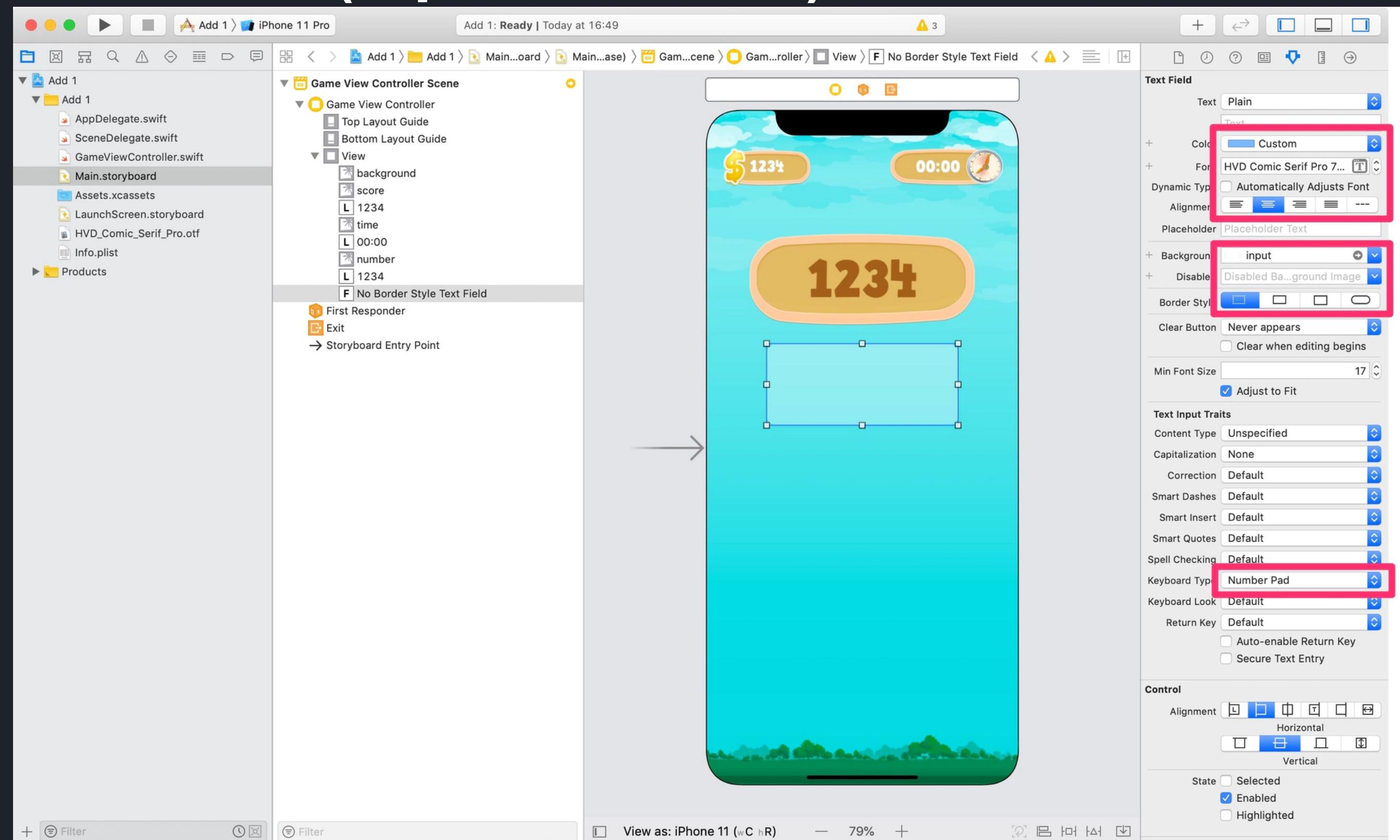
AspectFit, AspectFill, ScaleToFit

- **AspectFit** - Scales the content to the size of the view while ensuring that aspect ratio is maintained. Image may not fill the entire area (I.e. There may be white space).
- **AspectFill** - Scales content to the size of the view, while maintaining aspect ratio. Content may be clipped if it goes out of bounds.
- **ScaleToFit** - Scales the content to fit its own view, changing aspect ratio if necessary. Content will not be clipped.

Practice Time

- **ScoreImage** - Width: 120 | Height: 50
- **ScoreLabel** - Font Colour: White | Font Size: 22
- **TimeImage** - Width: 150 | Height: 50
- **TimeLabel** - Font Colour: White | Font Size: 22
- **NumberImage** - Width: 300 | Height: 130
- **NumberLabel** - Font Colour: #874F21 (135, 79, 33) | Font Size: 70
- **InputTextField** - Width: 250 | Height: 110 | Font Colour: #69A8FF (105, 168, 255) | Font Size: 70 |
Background: Input | Border Style: None | Keyboard Type : Number Pad (See Next Slide)
- **ExplanationLabel** - Width: 375 | Height: 65 | Font Colour: White | Font Size: 24
Label Text: "Add 1 to each of the digits. So, 1357 becomes 2468." | Lines: 0 | Text Alignment: Center

Practice Time (InputTextField)

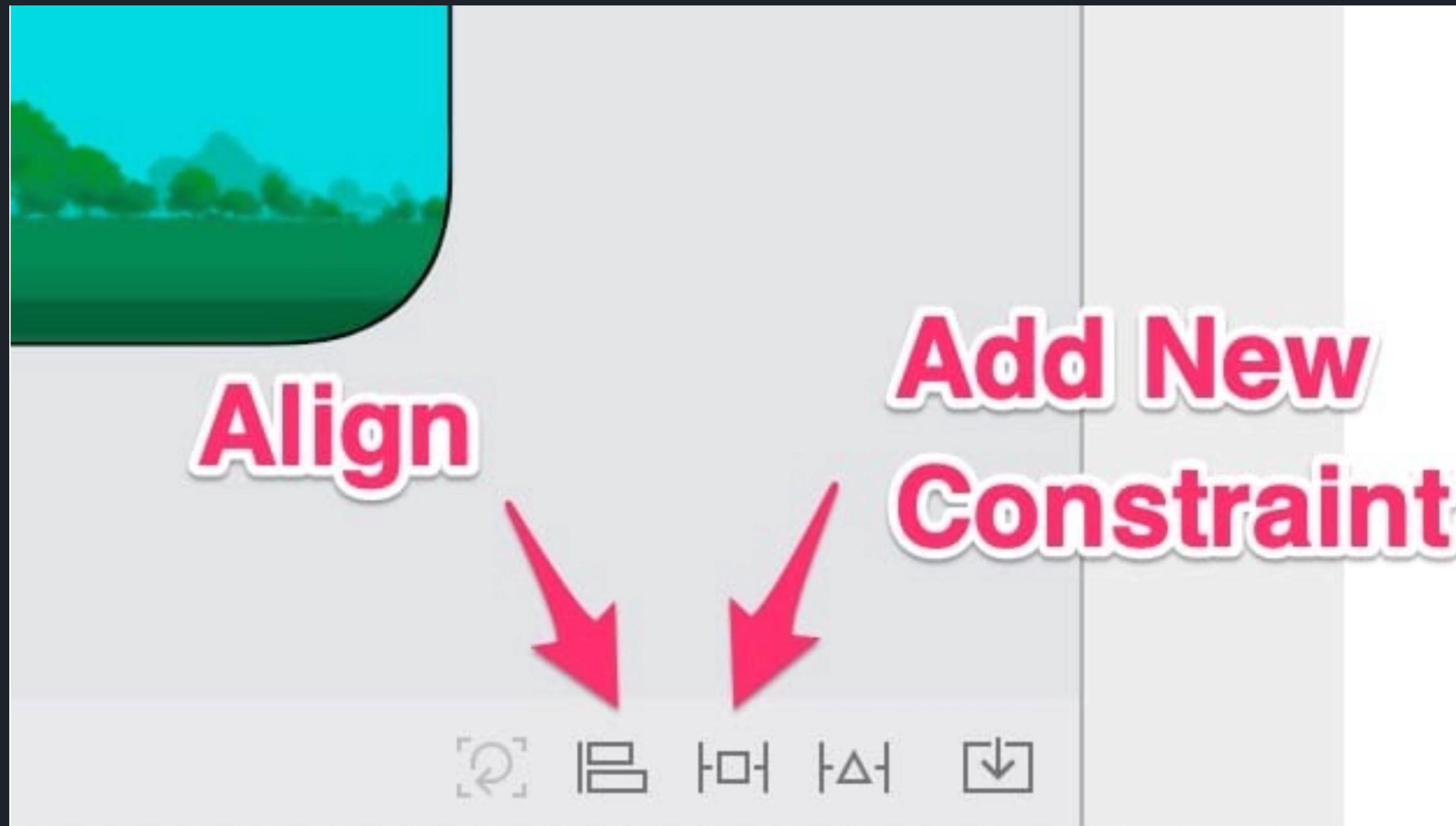


< / >

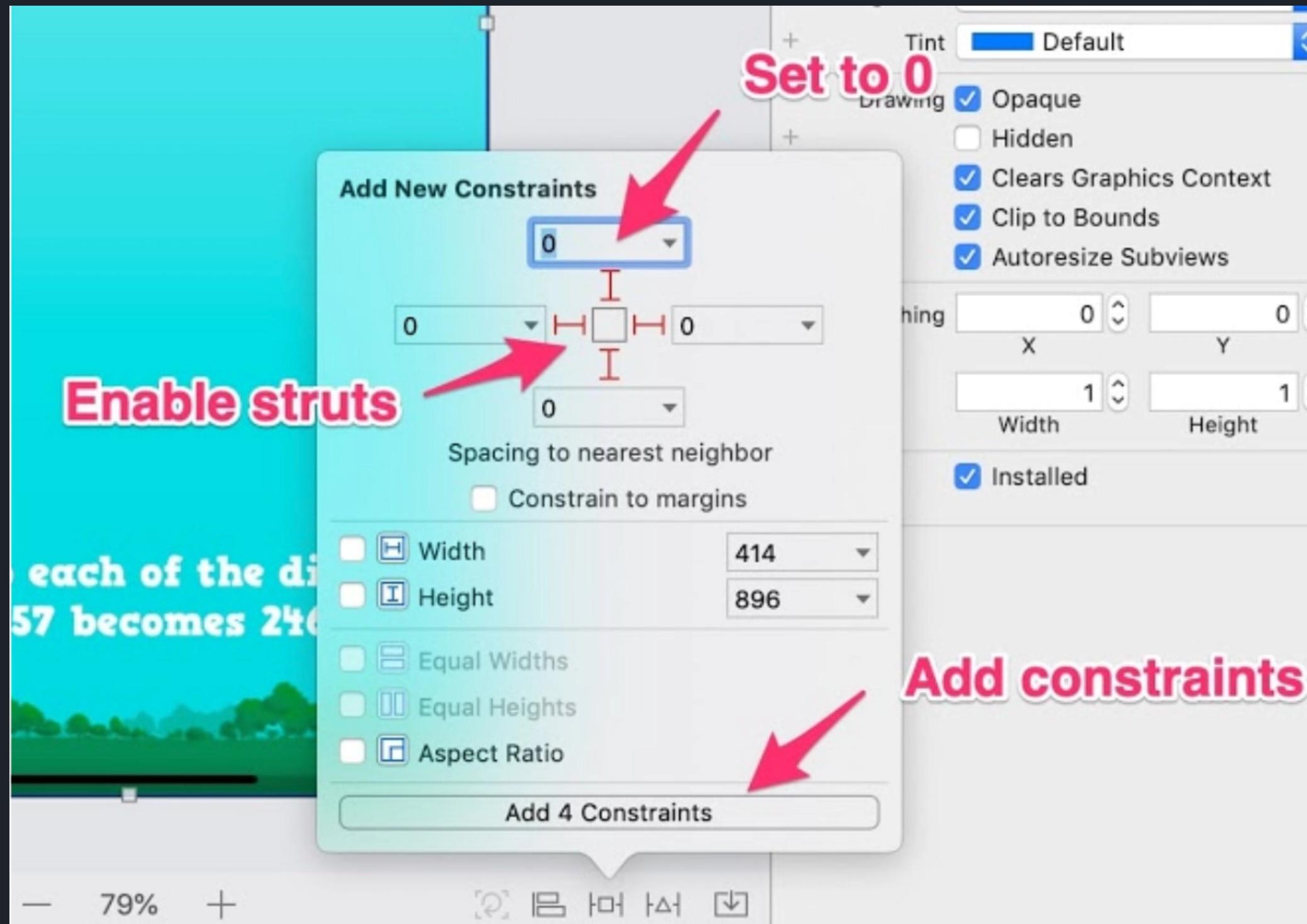
Auto-Layout

TO ENSURE THAT ELEMENTS' POSITIONS ARE DYNAMIC
(I.E. FIT NICELY REGARDLESS OF SCREEN SIZE)

Preview



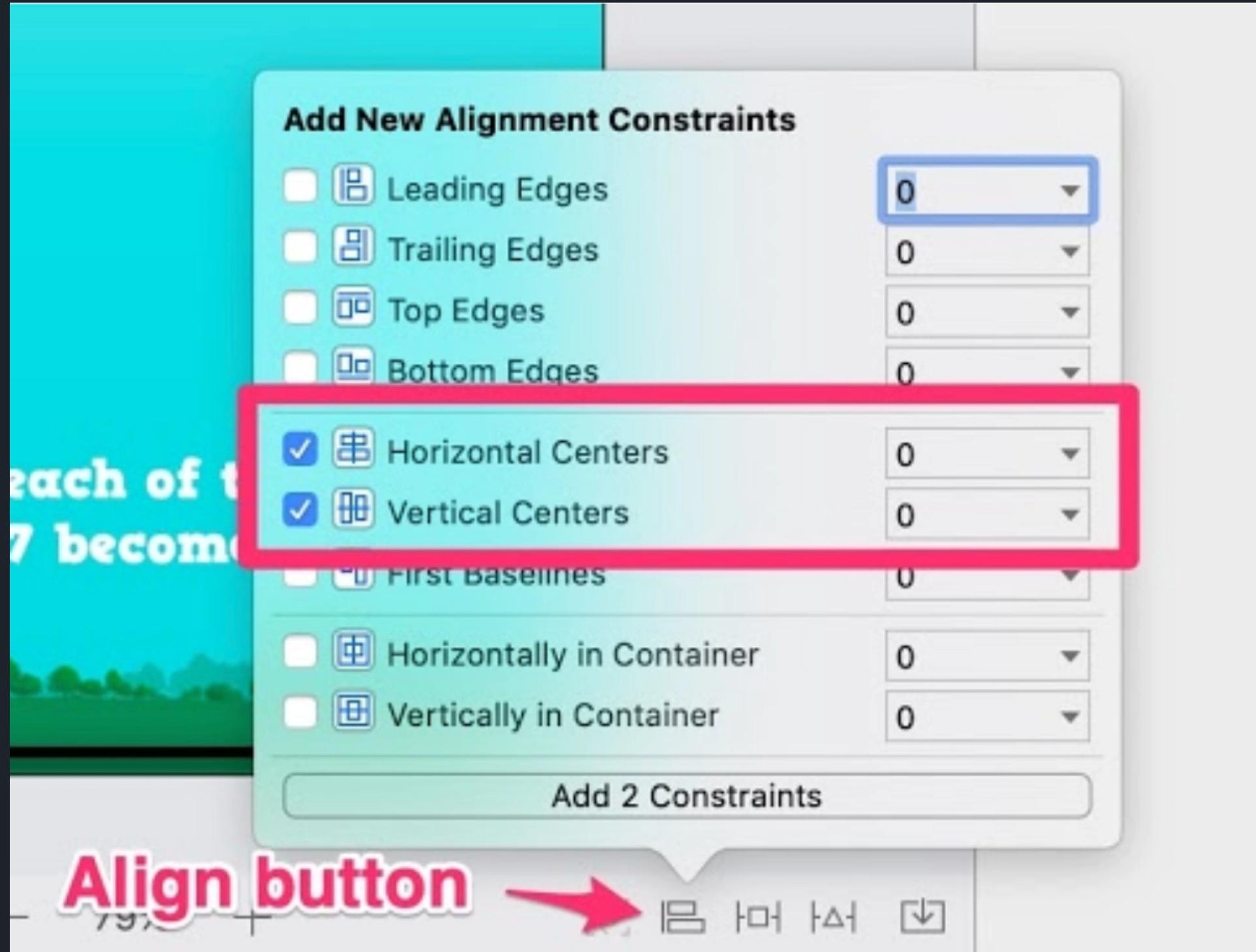
Constraints



Top, Bottom, Leading, Trailing

(Layman's Term) Setting **how far apart** an element should be from another element / edge of screen

Alignment



Top, Bottom, Leading, Trailing

(Layman's Term) Setting the
element's (x/y) position relative to
another

Recall that constraints only set the
distance apart from another
element, NOT the position

Practice Time

- **ScoreImage** - Top Constraint: 6 | Leading Constraint: 20
- **ScoreLabel** - Alignment: Horizontal, Vertical
- **TimeImage** - Trailing Constraint: 0 | Width: 150
- **TimeLabel** - Alignment: Horizontal, Vertical
- **NumberImage** - Alignment: Horizontal | Top: 60
- **NumberLabel** - Alignment: Horizontal, Vertical
- **InputField** - Top: 20 | Alignment: Horizontal
- **ExplanationLabel** - Bottom: 50 | Alignment: Horizontal

< / >

Outlets: Connecting Code with UI

ADD FUNCTIONALITY TO THE UI

Create Outlet Properties

```
@IBOutlet weak var scoreLabel:UILabel?  
@IBOutlet weak var inputField:UITextField?
```

Outlets provide a way to reference interface objects—the objects you added to your storyboard—from source code files.

- The `@IBOutlet` keyword indicates that we are creating an **outlet** property.
- `weak` - basically helps with memory issues and retain cycles

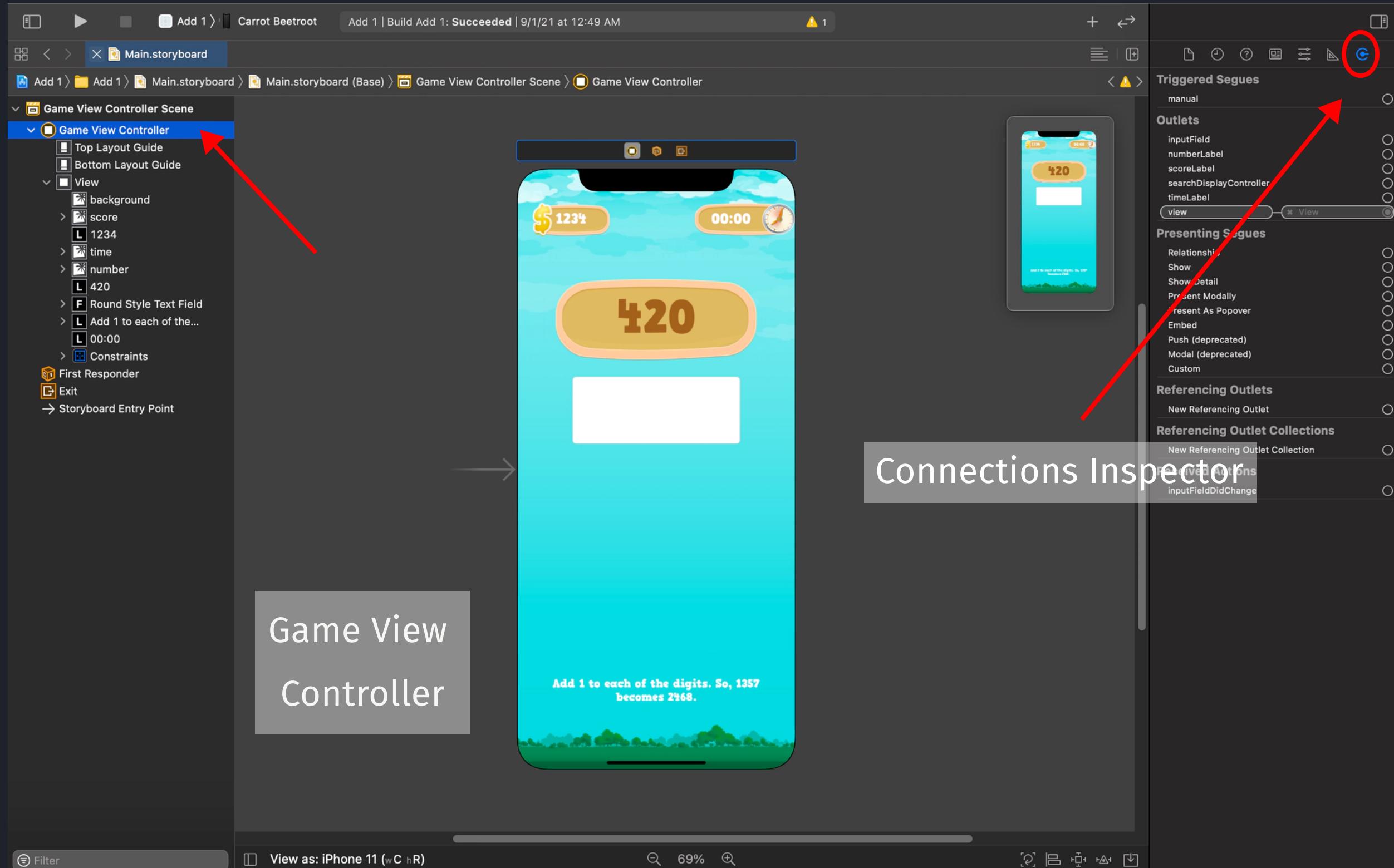
Now try creating the outlets for the timer label and number label!

But why are outlets optionals?

The value of an outlet cannot be set during initialization...When the view controller is instantiated, its view hasn't been loaded yet. It's only after the view controller is initialized that it loads its view. This also means that **any outlets declared in the view controller class don't have a value immediately after the view controller's initialization**. That's why an outlet is always declared as an (implicitly unwrapped) optional.

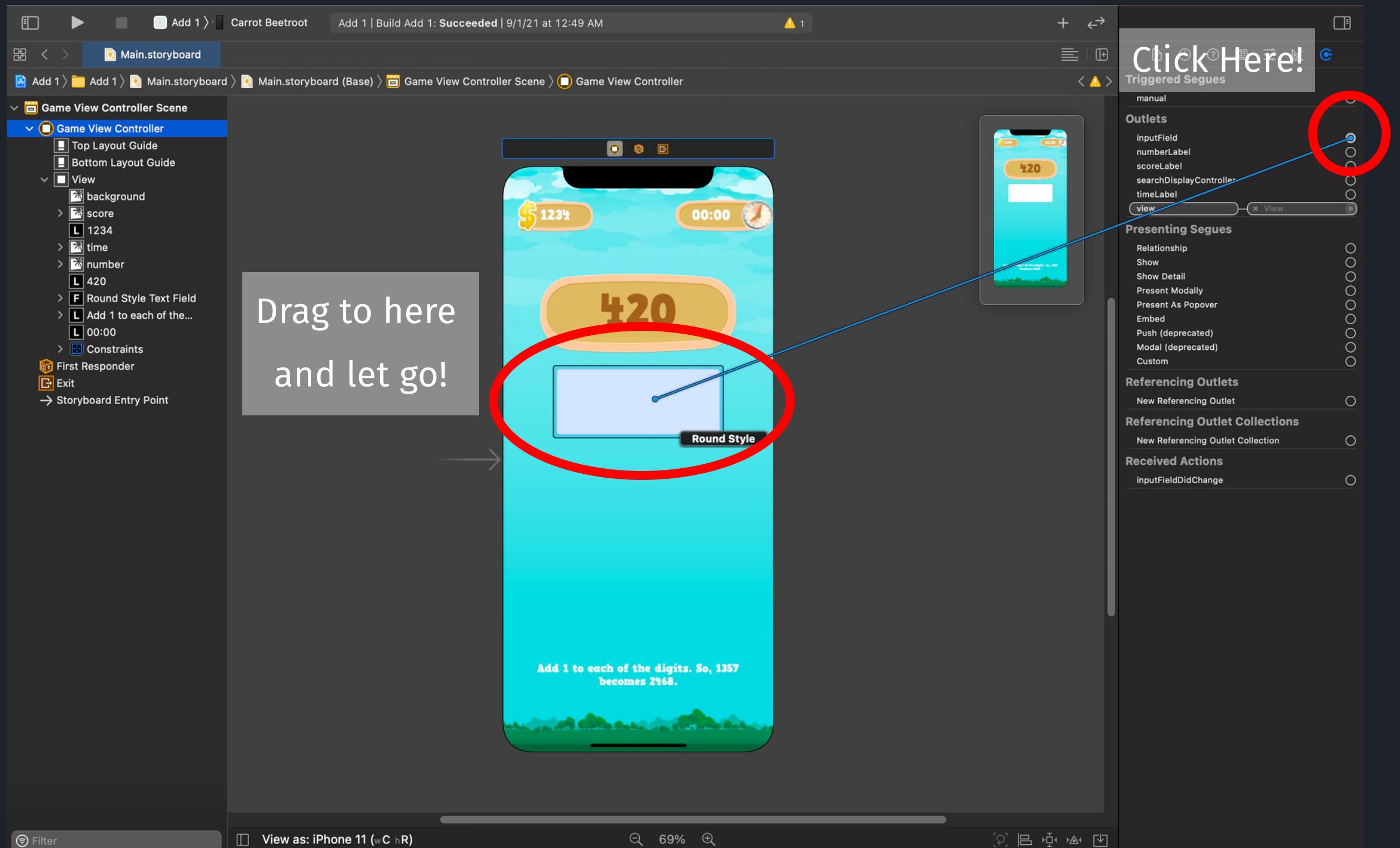
Source: <https://cocoacasts.com/should-outlets-be-optionals-or-implicitly-unwrapped-optionals>

Connect UI Elements to outlet properties



1. Open **Main.Storyboard**
2. Select **Game View Controller** in the Document Outline
3. Open **Connections Inspector** on the right

Make a Connection



1. Find the **inputField** property below Outlets
2. Click and drag from the circle on the right of the property to the **input field UI element** in the editor
3. Let go

< / >

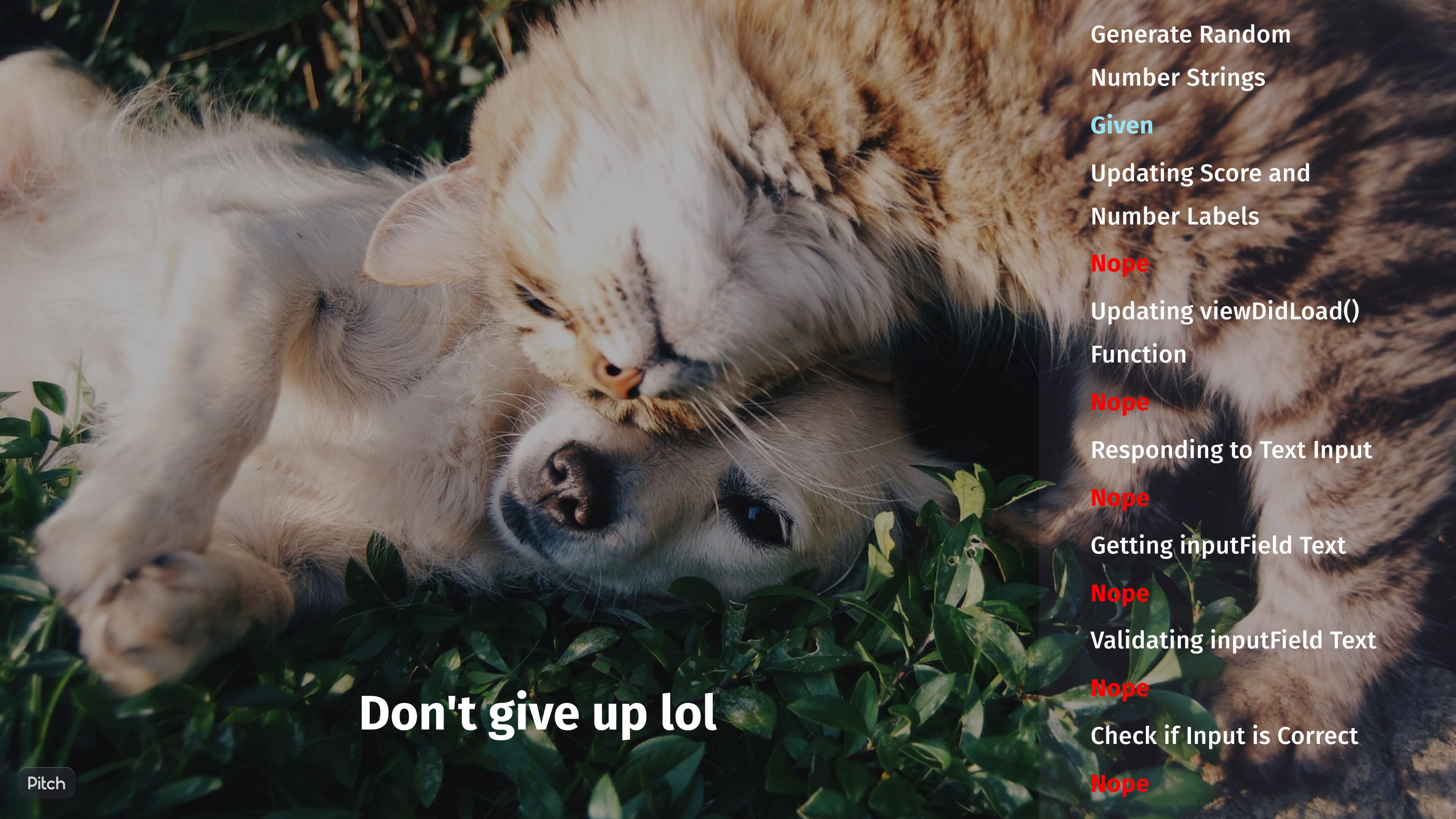
Repeat for the next 3 outlets!

(5 M I N S)

< / >

Code-Along! (Everything else)

HALFWAY THERE !



Generate Random Number Strings
Given
Updating Score and Number Labels
Nope
Updating viewDidLoad() Function
Nope
Responding to Text Input
Nope
Getting inputField Text
Nope
Validating inputField Text
Nope
Check if Input is Correct
Nope

Don't give up lol

Updating viewDidLoad() Function

Responding to Text Input

Getting inputField Text

Validating inputField Text

Check if Input is Correct

thats it..?

This is a slide with tons of text

THE CORE OF YOUR IDEA SUMMARIZED IN LESS THAN 10 WORDS

Expand on your idea. Tell why it matters, and why the listeners should care. Be informative, but also be interesting. Tell the story in a way that would resonate with your audience. Don't be afraid to throw in a joke, an unexpected metaphor, or a silly pun.

ANOTHER IDEA, EquALLY IMPORTANT

A presentation is the process of presenting a topic to an audience. It is typically a demonstration, introduction, lecture, or speech meant to inform, persuade, inspire, motivate, or to build good will or to present a new idea or product.

A code snippet

```
body {  
  font-family: Arial, Helvetica, sans-serif;  
}
```

The `font-family` property defines the font that is applied to the selected element. The font that is selected is not a single font face, but a "family", and thus may be dependent on other typographic property values to select the correct face within the family.

A value can be one of the following:

- A font family name that matches a font that is embedded on the page or available on the user's system.
- A series of family names, separated by commas, which can include a generic family name

Source: css-tricks.com

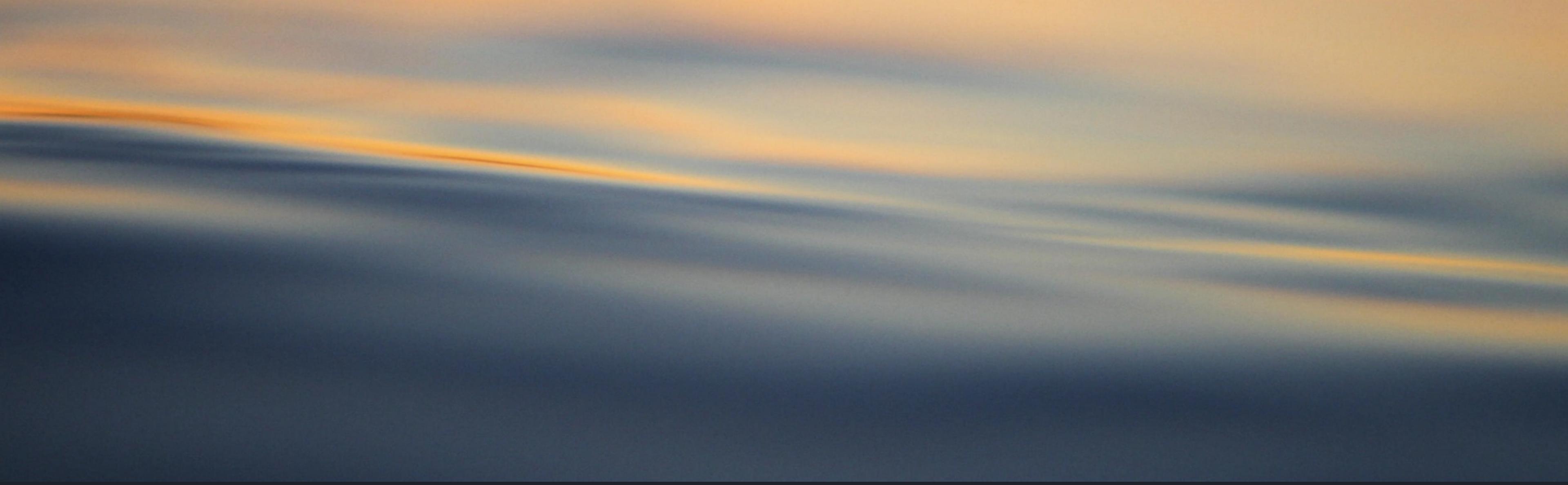
This is a headline

THIS IS A SUBHEADLINE

This is space for you to expand. Just start typing away.

Write as much as you want (or as little, if you're channeling Hemingway). A little Lorem ipsum to get your creative juices flowing: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi rutrum consectetur dolor pulvinar pharetra.





Make your point visually

This is space for you to expand. Just start typing away. Write as much as you want (or as little, if you're channeling Hemingway). A little Lorem ipsum to get your creative juices flowing: *Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi rutrum consectetur dolor pulvinar pharetra.*



A great photo deserves a great caption, don't you think?

DESCRIBE WHAT THIS SLIDE IS ABOUT



This is an important point

Expand on it here. Why is it important?

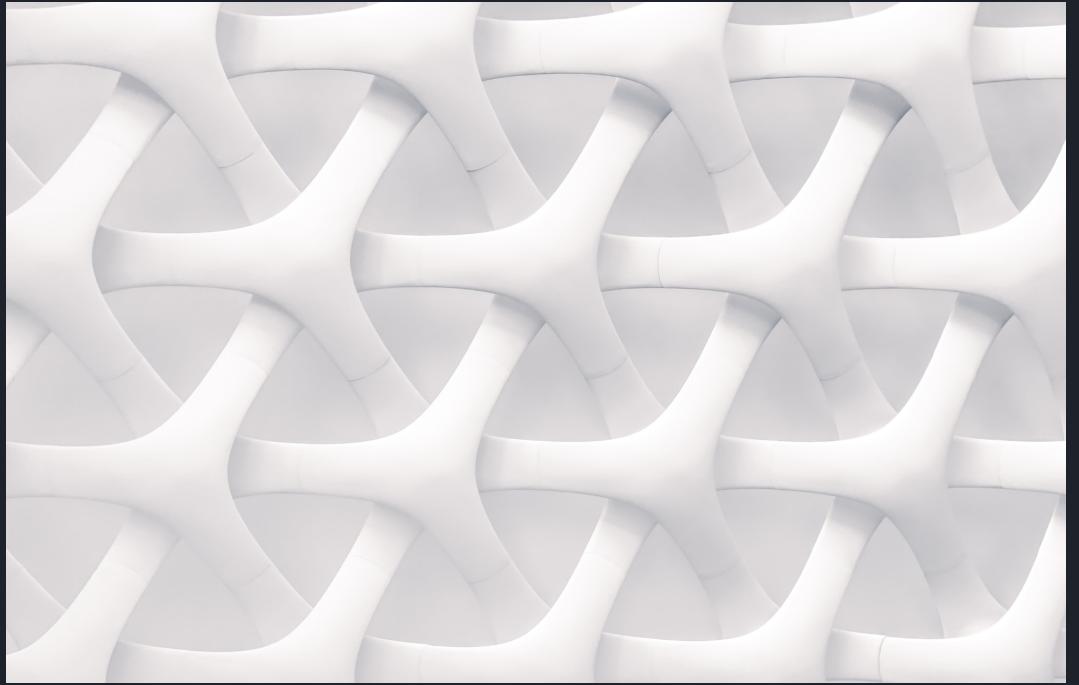
Why does it matter?

Another important point

You already know that it's important.

But what about your listeners?

DESCRIBE WHAT THIS SLIDE IS ABOUT



This is an important point

Expand on it here. Why is it important?

Why does it matter?

Another important point

You already know that it's important.

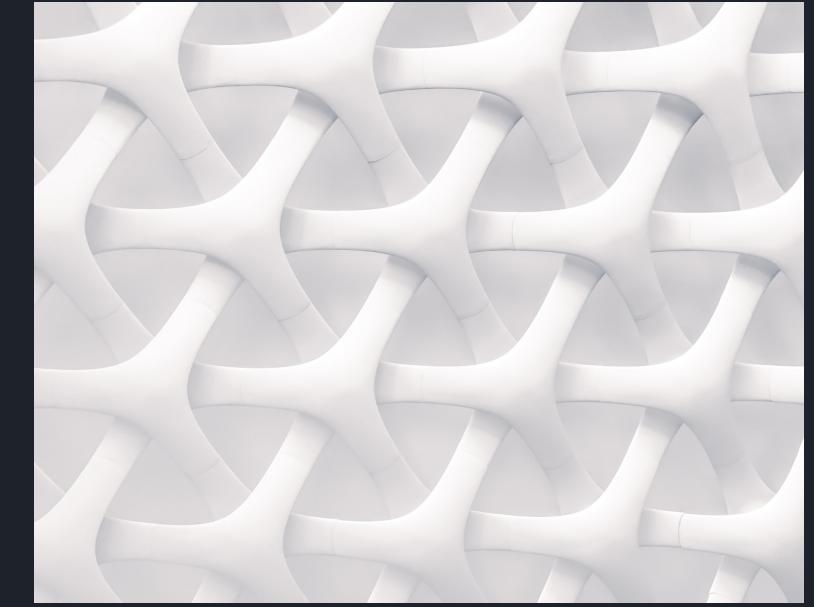
But what about your listeners?

They're all important points!

Convince the audience, both with facts

and with GIFs.

DESCRIBE WHAT THIS SLIDE IS ABOUT



This is an important point

Expand on it here. Why is it important? Why does it matter?

Another important point

You already know that it's important. But what about your listeners?

Another important point

Write as much as you want (or as little, if you're channeling Hemingway).

They're all important points!

Convince the audience, both with facts and with GIFs.



A list about lists

- Lists are a great communication tool
- They don't require as much focus and concentration as full paragraphs
- They are much easier to scan
- In a way, they're much closer to bursts of real speech
- Use lists when you believe that your readers don't have much time and can't really focus and read
- Okay, that's the end of the list

< / >

**The main problem with quotes on the internet
is that sometimes they're not true.**

ANONYMOUS GHOSTWRITER

THIS IS A VIDEO SLIDE



And here is a caption for this video

**Last but not least,
we need to code the
countdown timer!**

**This is really the last part,
we promise!**

USE THIS SLIDE TO TALK ABOUT TIME



Thing that happened first

Here you can just say stuff in
no particular order

Second thing

Instead of full sentences
You can just list stuff
Like it's a list
Of things

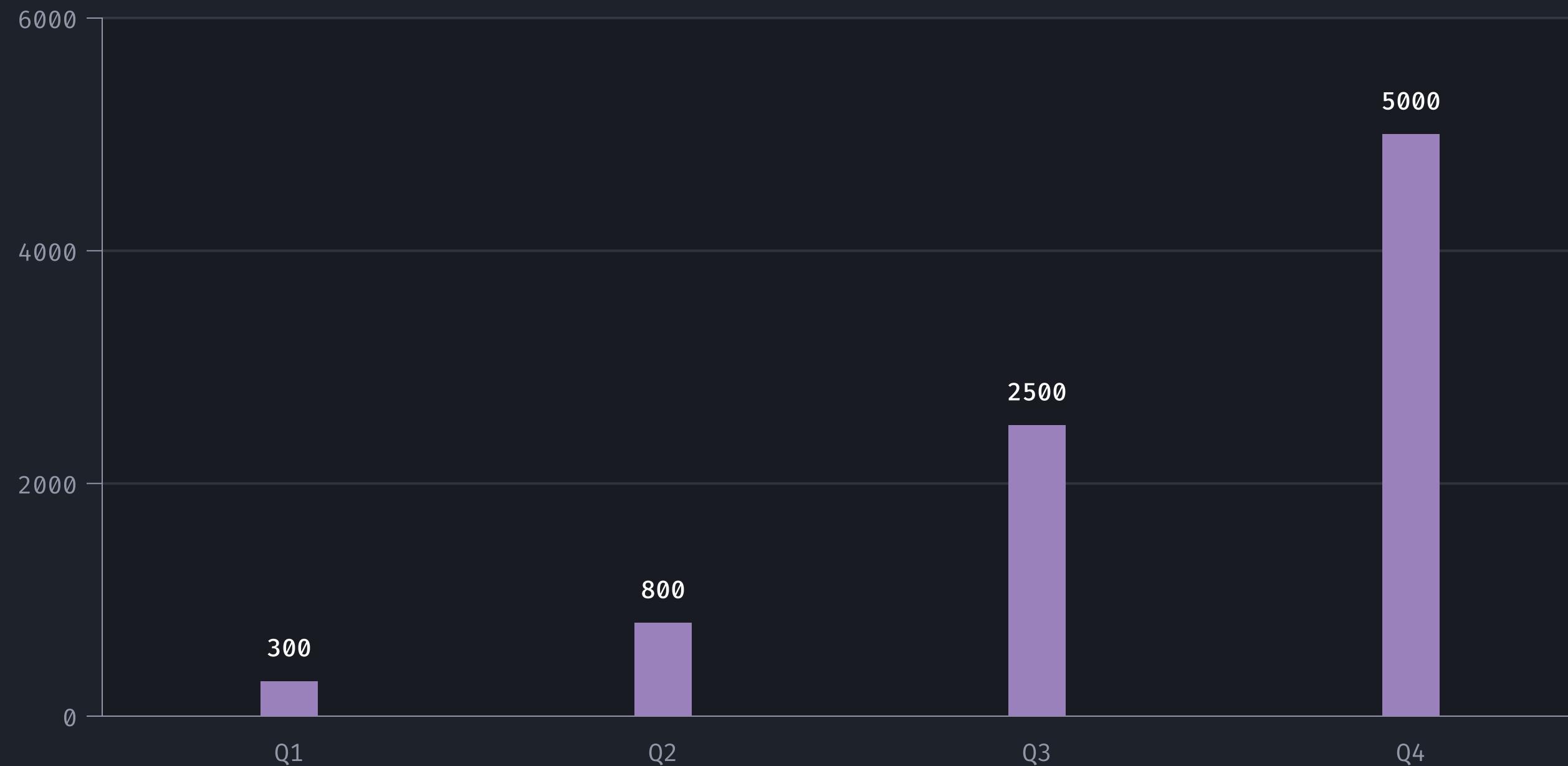
Third thing, maybe Q3

Add a goal here
And another goal
And another one

Thing that happened last

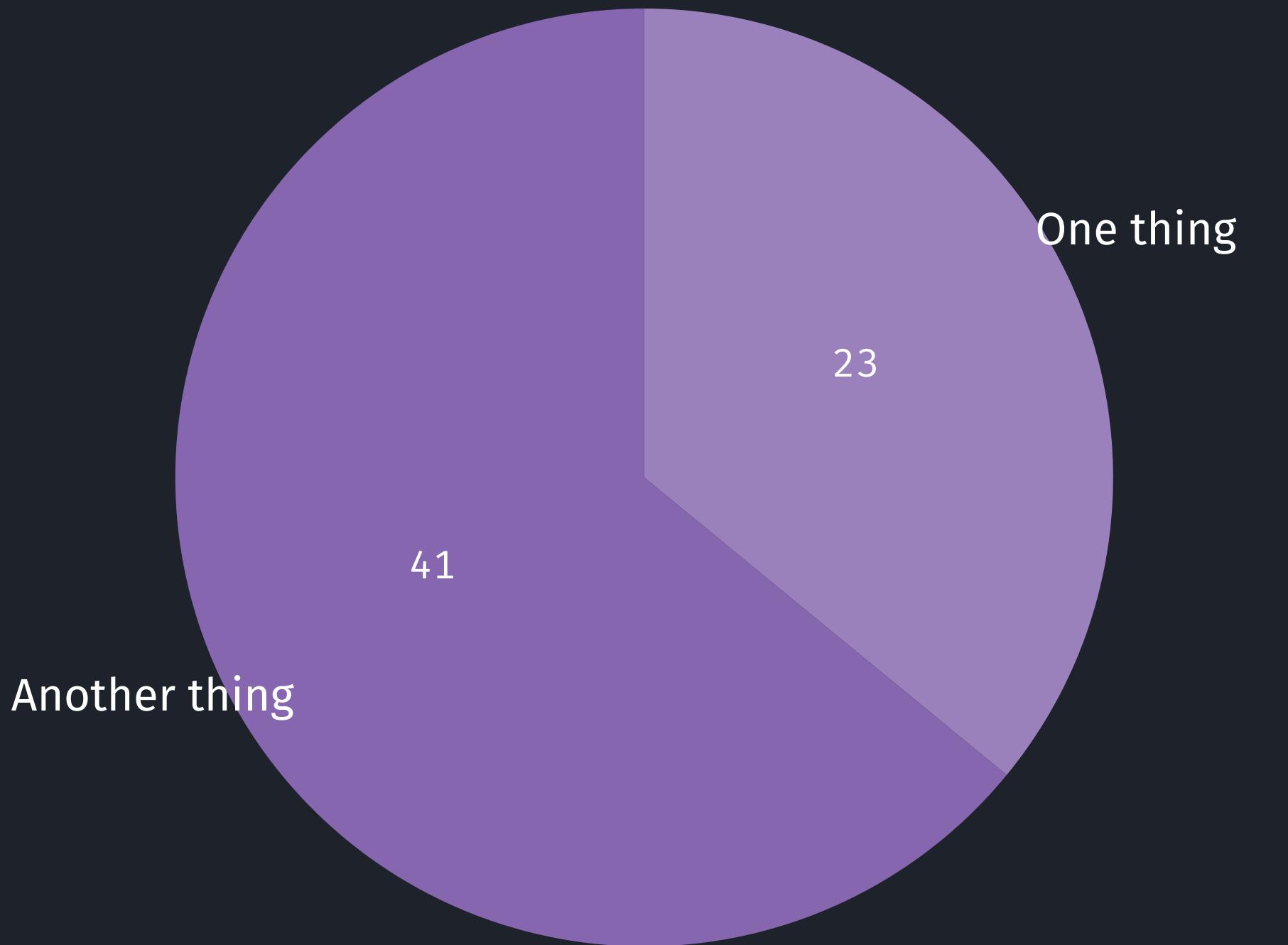
Offer a summary or conclusion,
if possible

GIVE YOUR GRAPH A NAME



Explain the meaning of the graph and help people understand what to look at

THIS IS A PIE CHART SLIDE



About the data

Some background information about the graph: where the numbers come from, what they mean, and why it matters.

THIS IS A TABLE

| Name | Role | Location | Spirit Animal |
|--------|-----------|-------------|--|
| Simon | Sales | Germany |  Sloth |
| Diego | Design | America |  Rhino |
| Mae | Marketing | Singapore |  Flamingo |
| Fatima | Founder | New Zealand |  Unicorn |

And here is a caption for this table

CONTACT

Address

Präsentationstr. 7,
72013 Slide-am-Deck, Gerone,
Planet Mars

Phone

+1 800 314 159 265

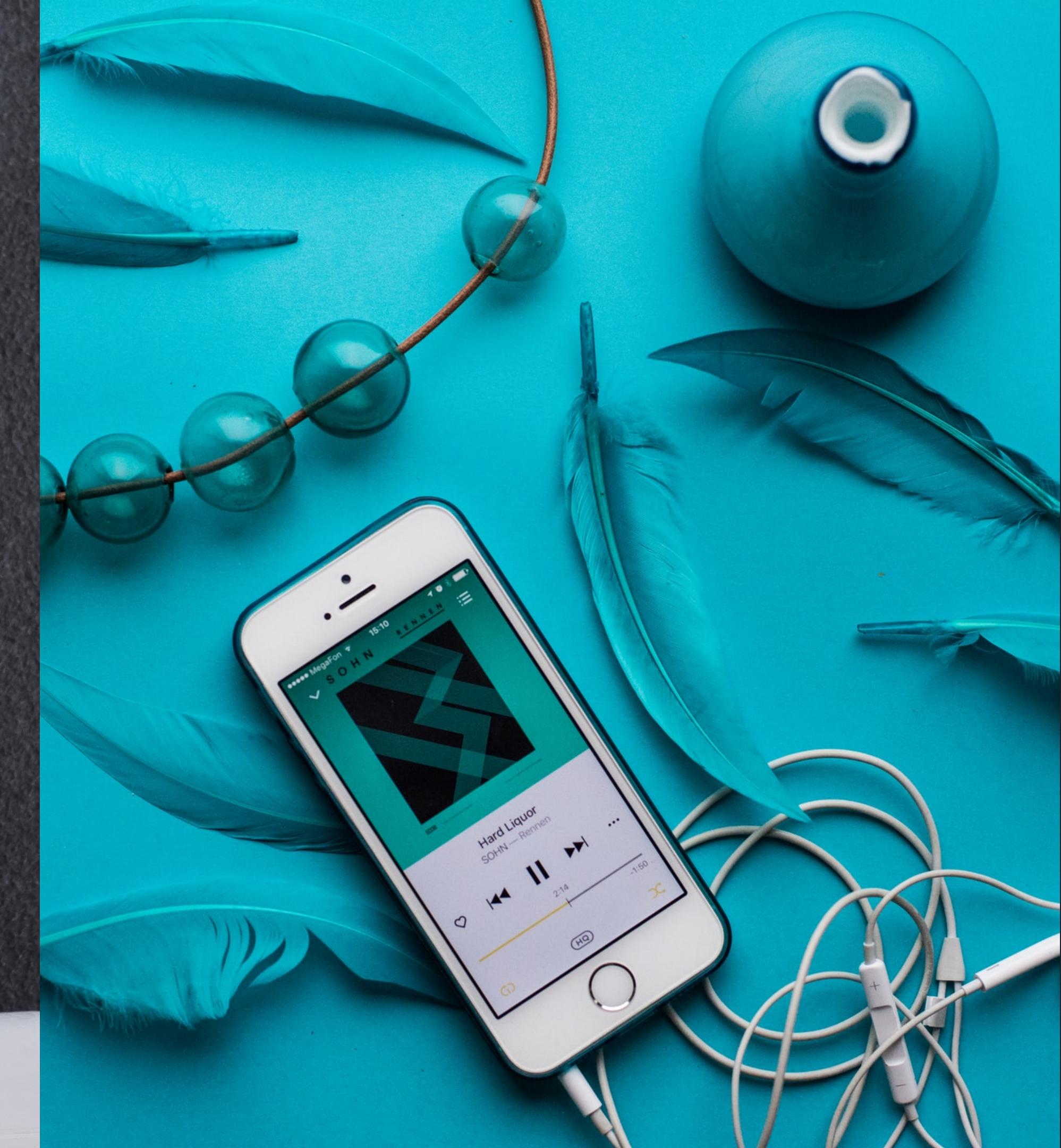
Email

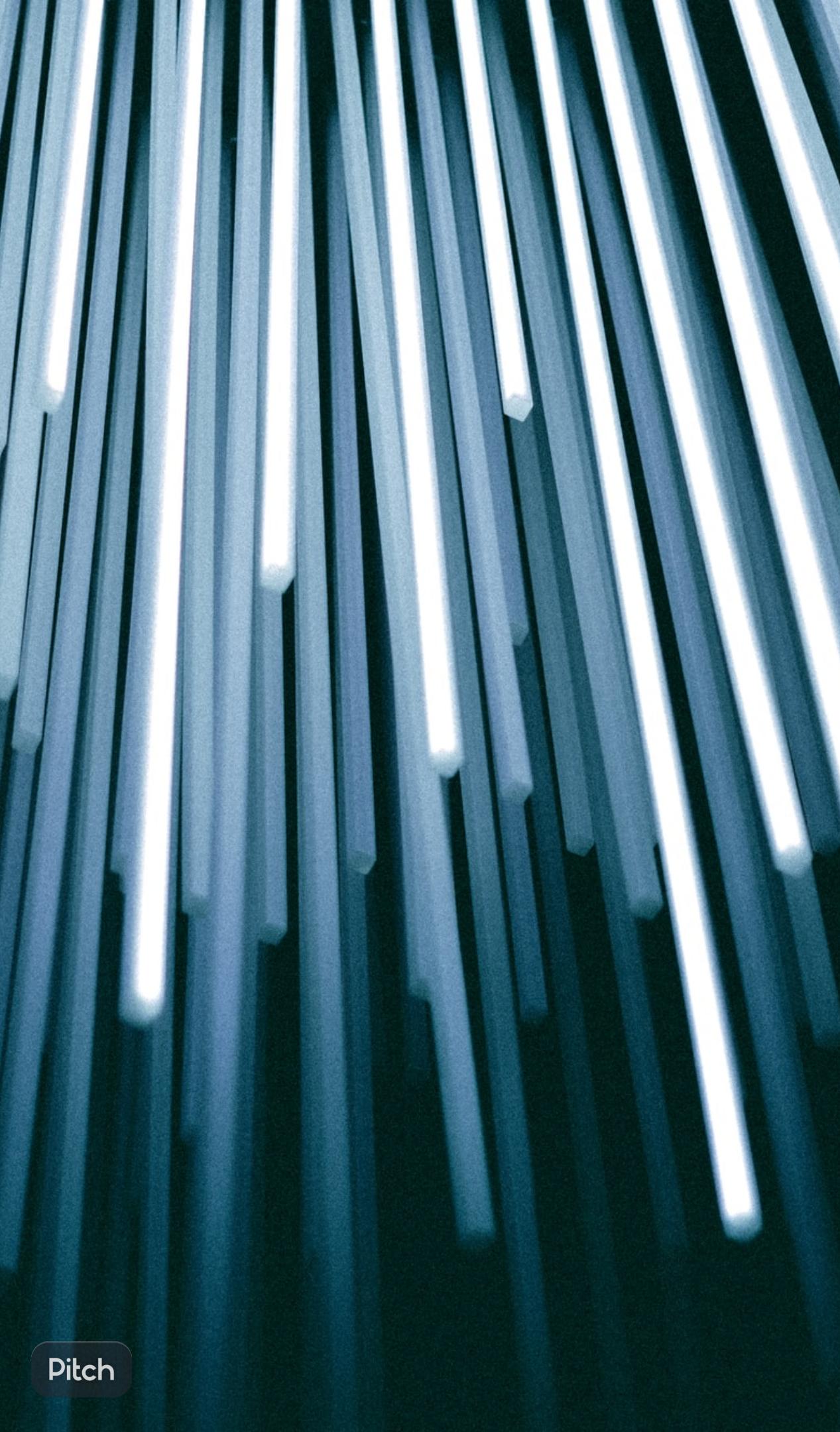
very@nice.email



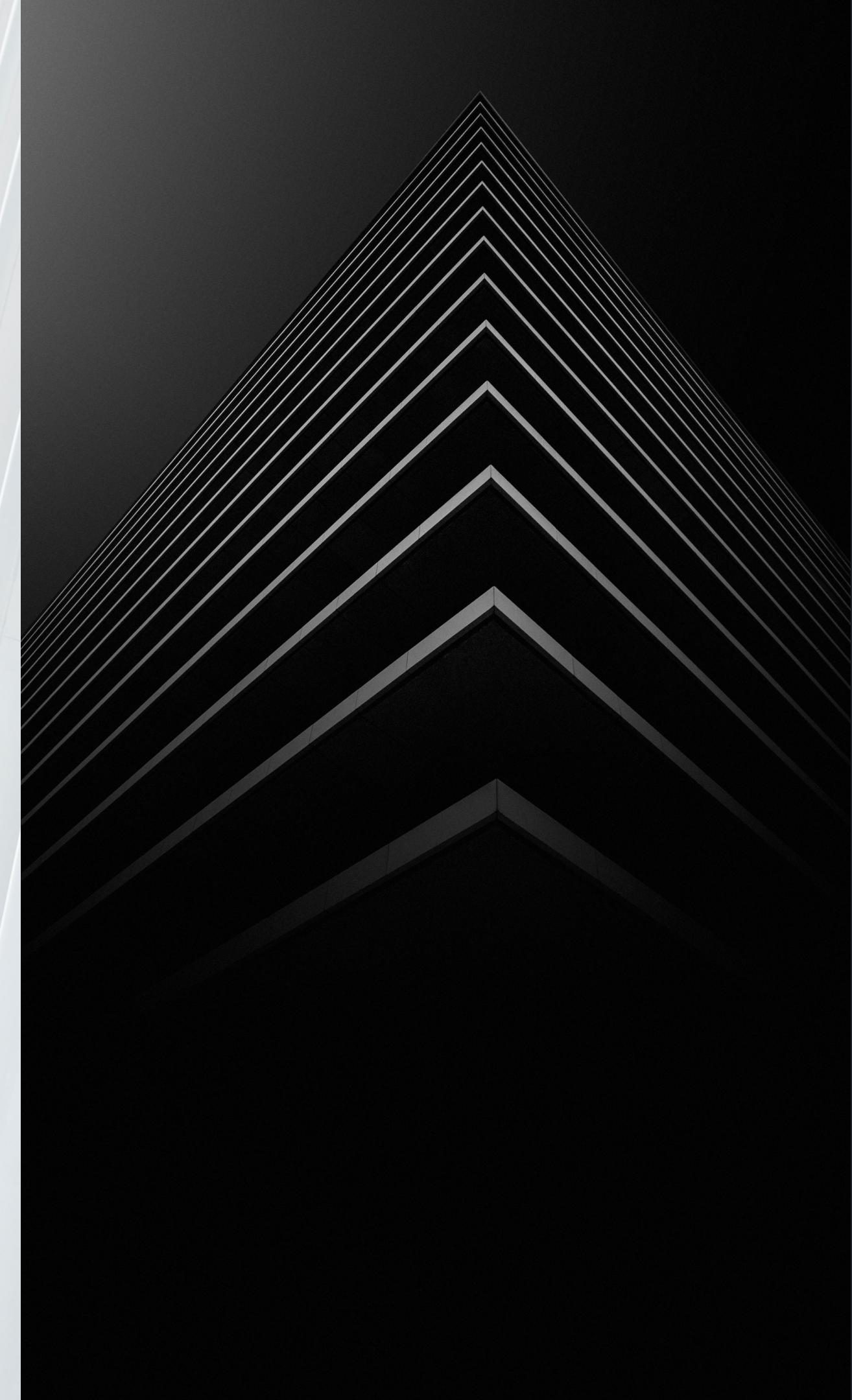


Pitch





Pitch





Pitch

