移动平台开发技术

iOS开发中的UI进阶

章国锋 2025春夏学期

浙江大学
ZHEJIANG UNIVERSITY

如何完善你的App

深入理解SwiftUI中的交互设计

复杂动画的设计与实现

浙江大学
ZHEJIANG UNIVERSITY

如何完善你的App

深入理解SwiftUI中的交互设计

复杂动画的设计与实现

如何完善你的App

Tips for perfecting your app

浙江大學
ZHEJIANG UNIVERSITY

优化Onboarding体验

如何完善你的App

Tips for perfecting your app

繁琐的注册流程

照片权限请求 推送权限请求

各种权限请求

相机权限请求 蓝牙权限请求

Onboarding

复杂的操作流程

无法跳过的开屏广告

如何完善你的App

Tips for perfecting your app

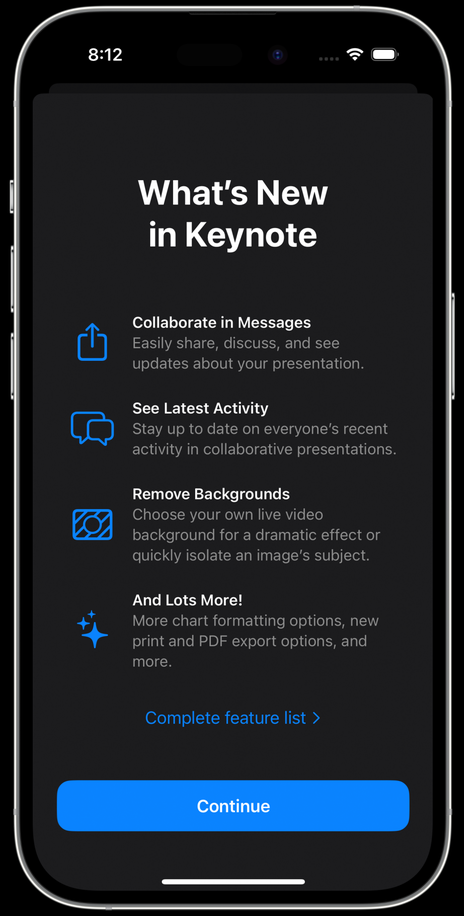
浙江大学 ZHEJIANG UNIVERSITY

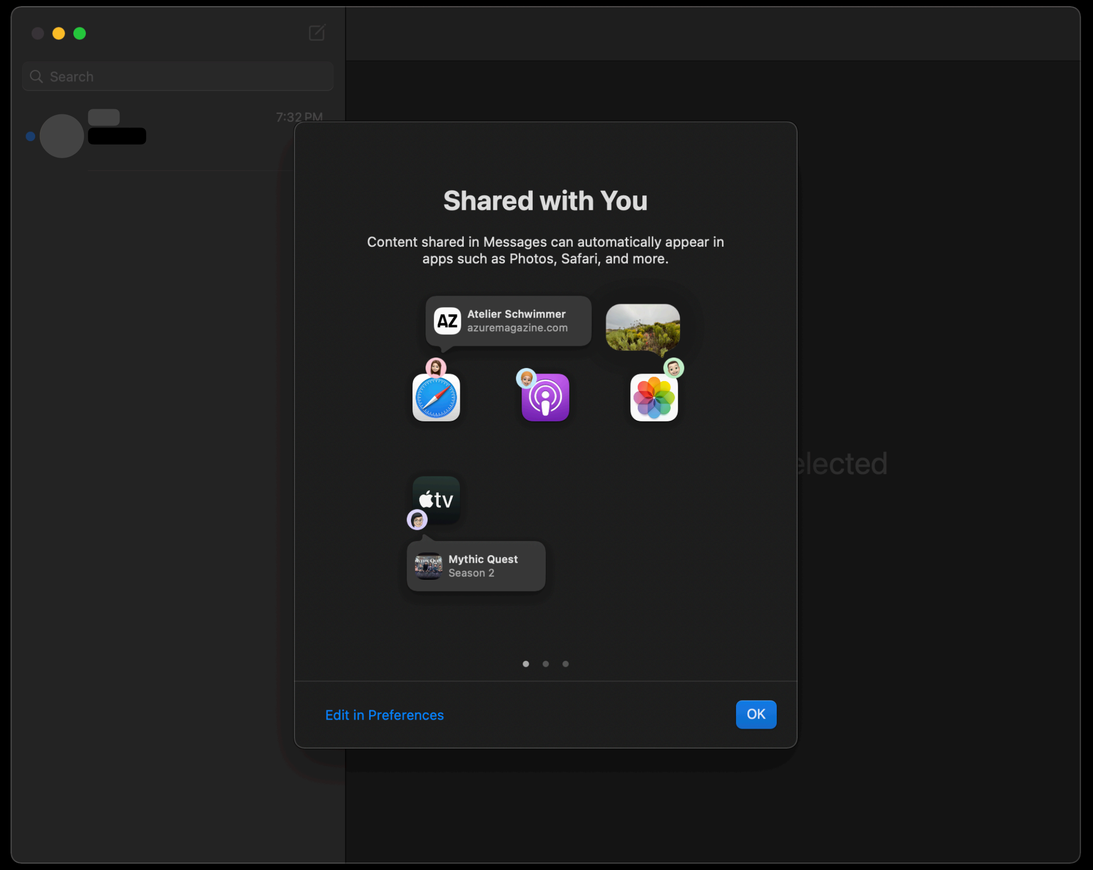
优化Onboarding体验

1. 介绍App更新的新功能、新特性
2. 因App需要，必须注册才能使用，引导用户注册
3. 通过用户选择进行个性化App初始化
4. 通过引导让用户快速熟悉App使用方法

如何完善你的App

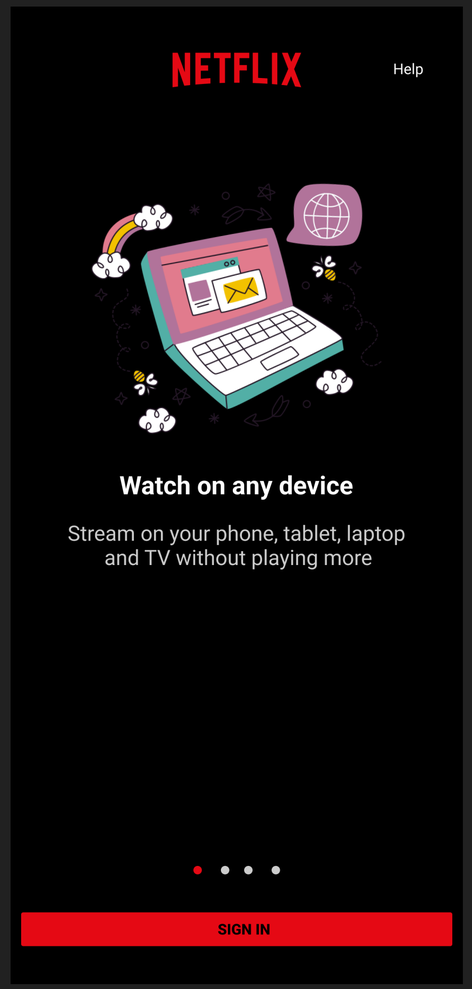
Tips for perfecting your app

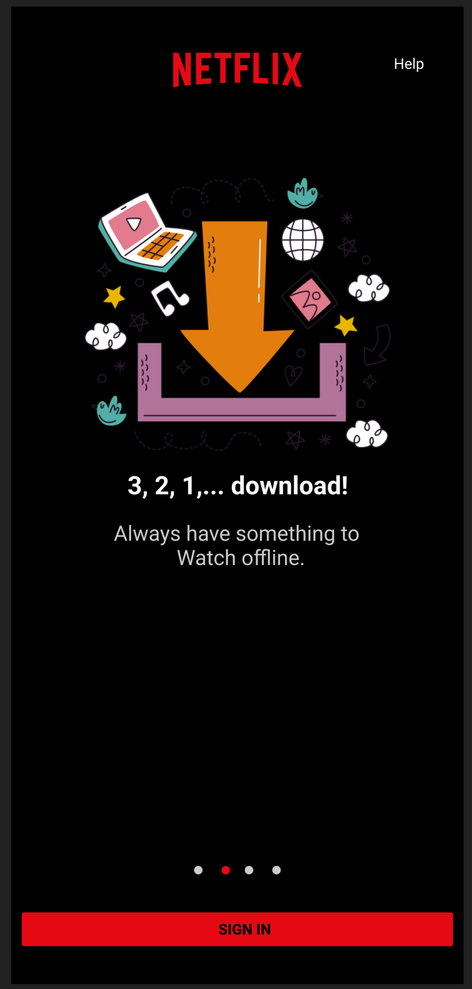


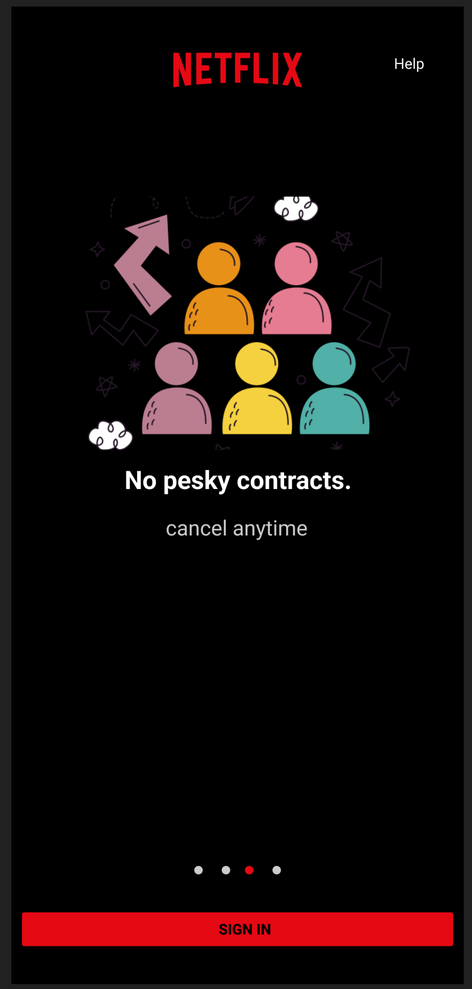


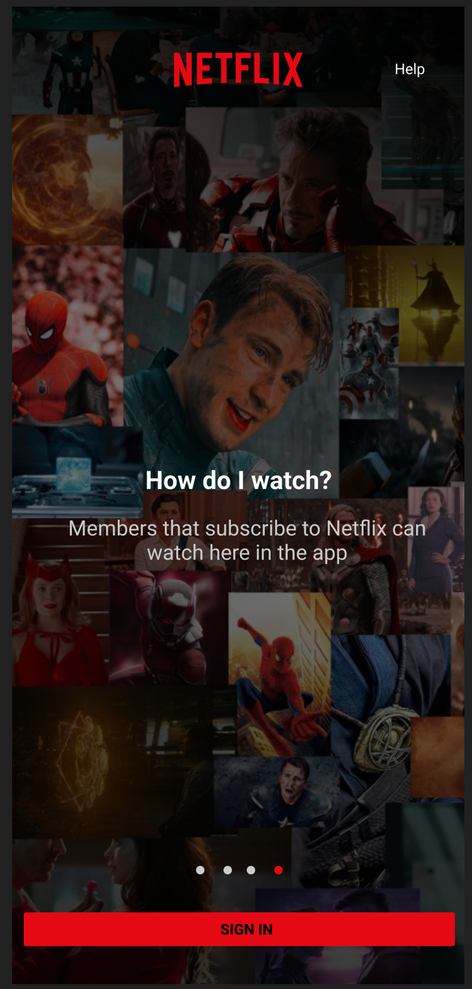
如何完善你的App

Tips for perfecting your app



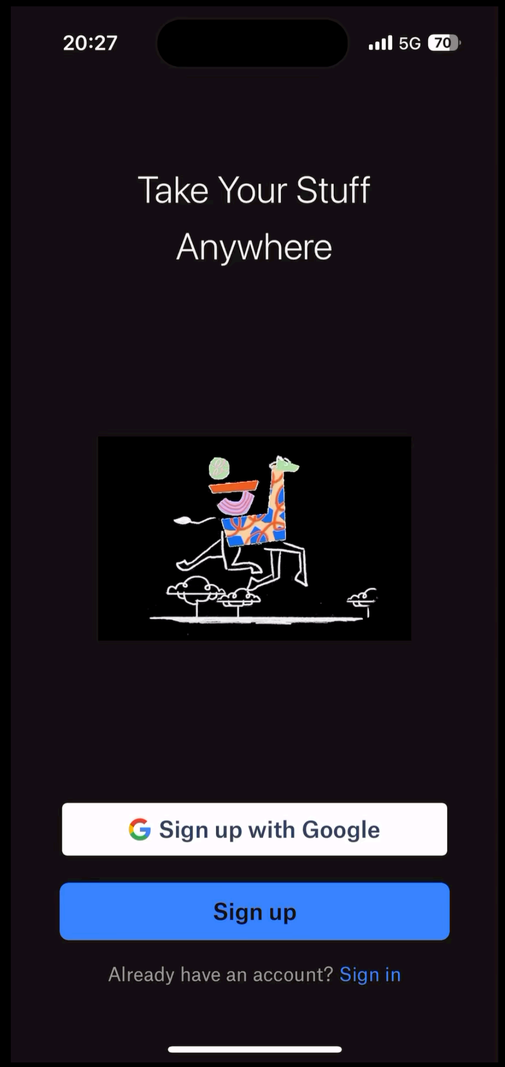






如何完善你的App

Tips for perfecting your app

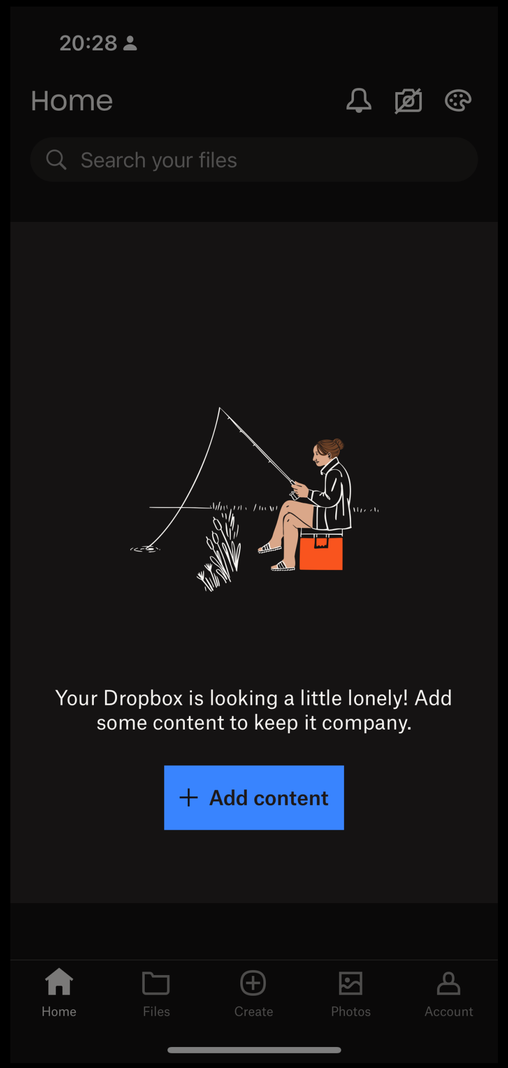


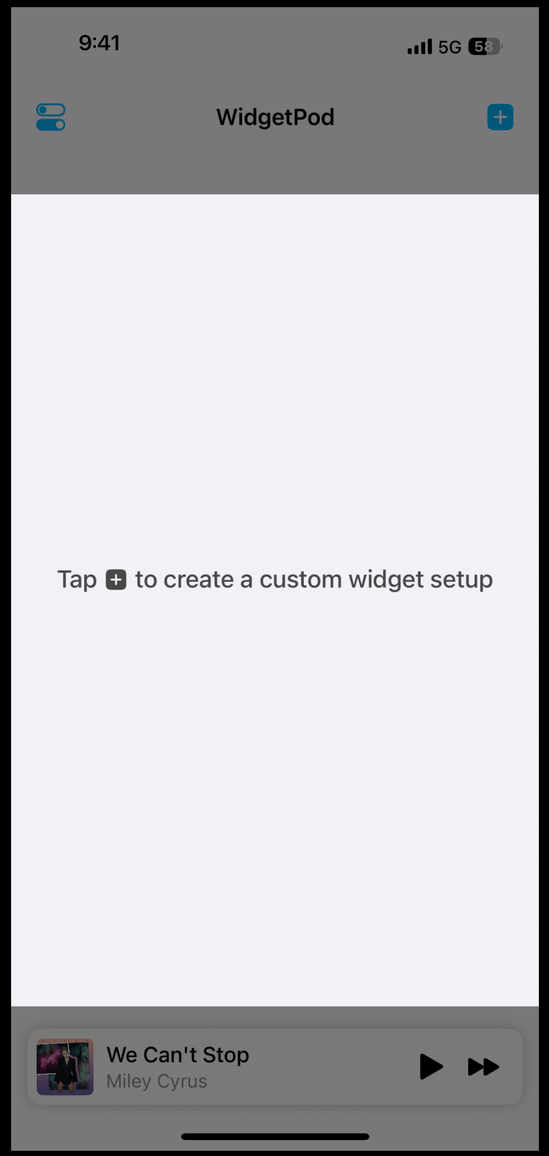
浙江大学 ZHEJIANG UNIVERSITY

如何完善你的App

Tips for perfecting your app

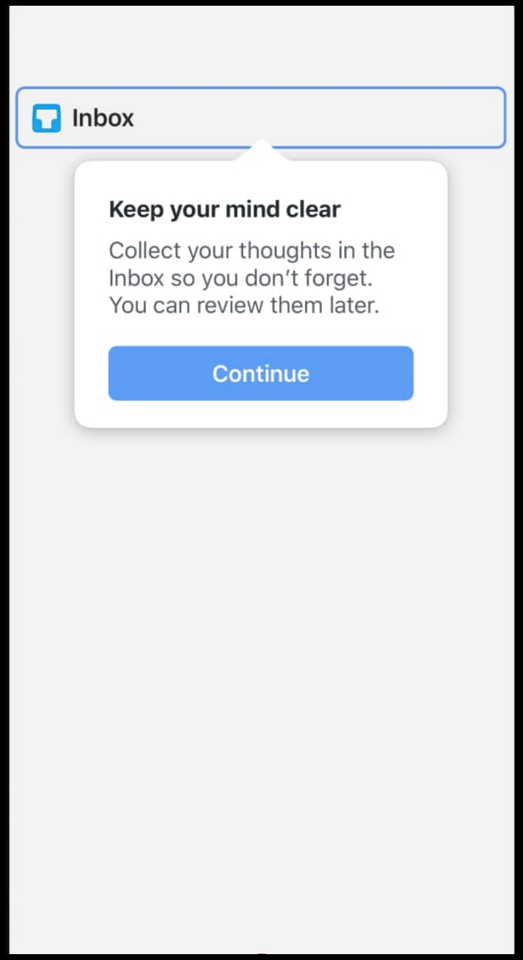
浙江大学 ZHEJIANG UNIVERSITY

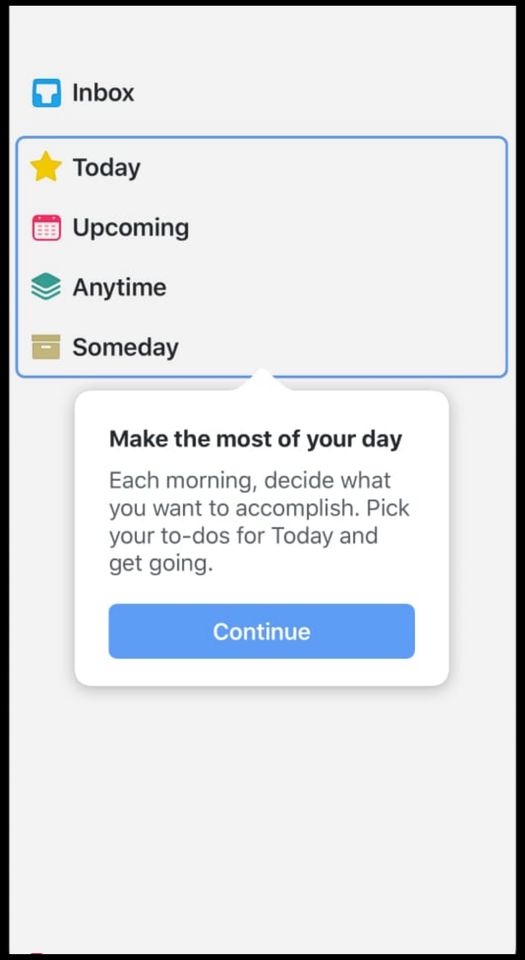




如何完善你的App

Tips for perfecting your app

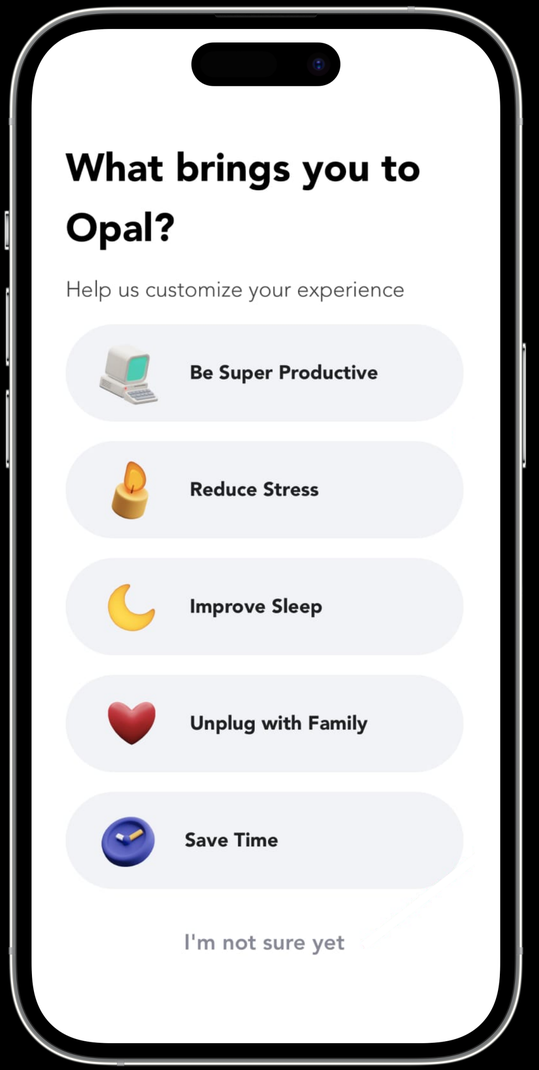




如何完善你的App

Tips for perfecting your app

1. 引导页介绍新功能或引导用户注册
2. 使用Sheet或至少有一个跳过按钮
3. 减少不必要的选择和信息输入



如何完善你的App

Tips for perfecting your app

在开发之初多使用现有工具

如何完善你的App

Tips for perfecting your app

浙江大学 ZHEJIANG UNIVERSITY

SF Symbol

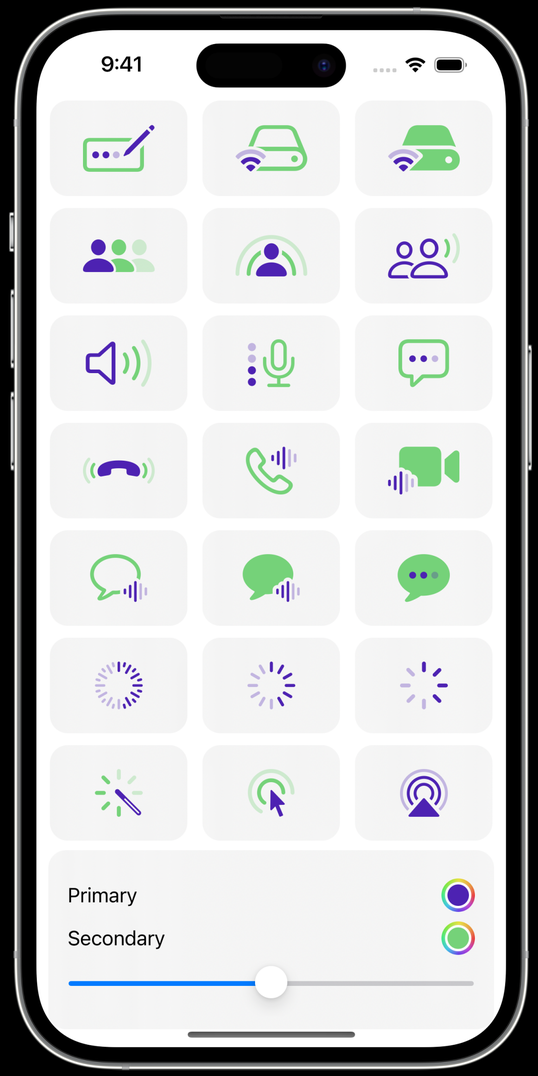
如何完善你的App

Tips for perfecting your app

浙江大学 ZHEJIANG UNIVERSITY

SF Symbol

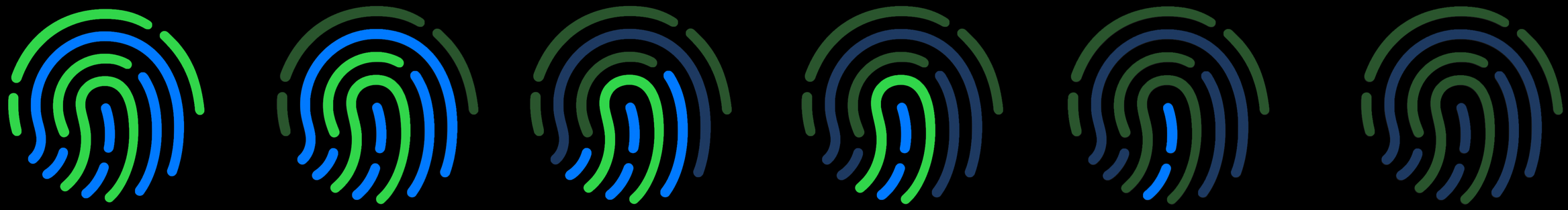
- 目前有超过 12, 000个SF symbols
- 支持使用不同的颜色
- 在不同rendering 模式下可以高亮部分区域



如何完善你的App

Tips for perfecting your app

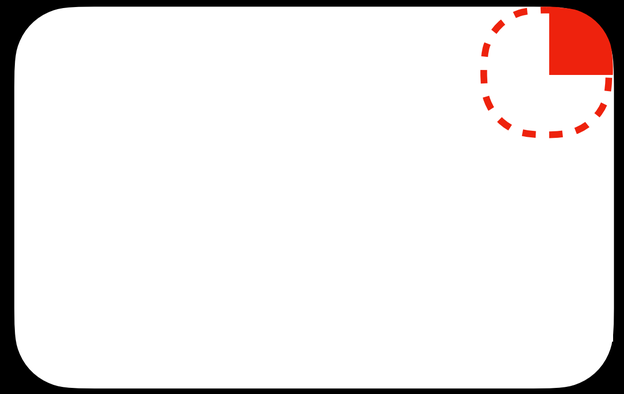
浙江大学 ZHEJIANG UNIVERSITY



```
Image(systemName: "touchid", variableValue: value)
    .symbolRenderingMode(.palette)
    .foregroundStyle(.blue, .green)
```

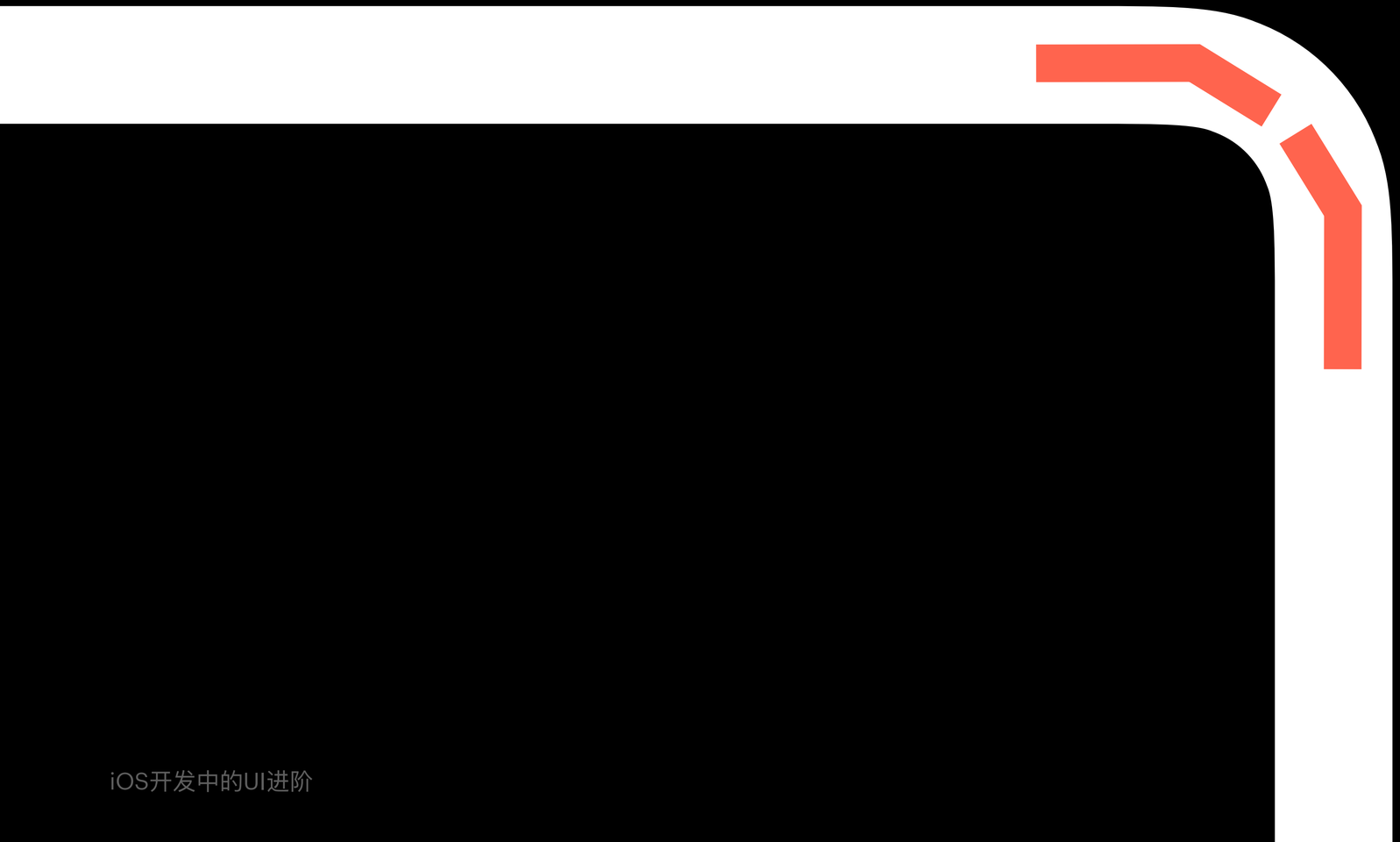
如何完善你的App

Tips for perfecting your app



如何完善你的App

Tips for perfecting your app



浙江大学 ZHEJIANG UNIVERSITY

如何完善你的App

Tips for perfecting your app

浙江大学 ZHEJIANG UNIVERSITY

如何完善你的App

Tips for perfecting your app


```
RoundedRectangle(cornerRadius: 30, style: .continuous)
```

如何完善你的App

Tips for perfecting your app

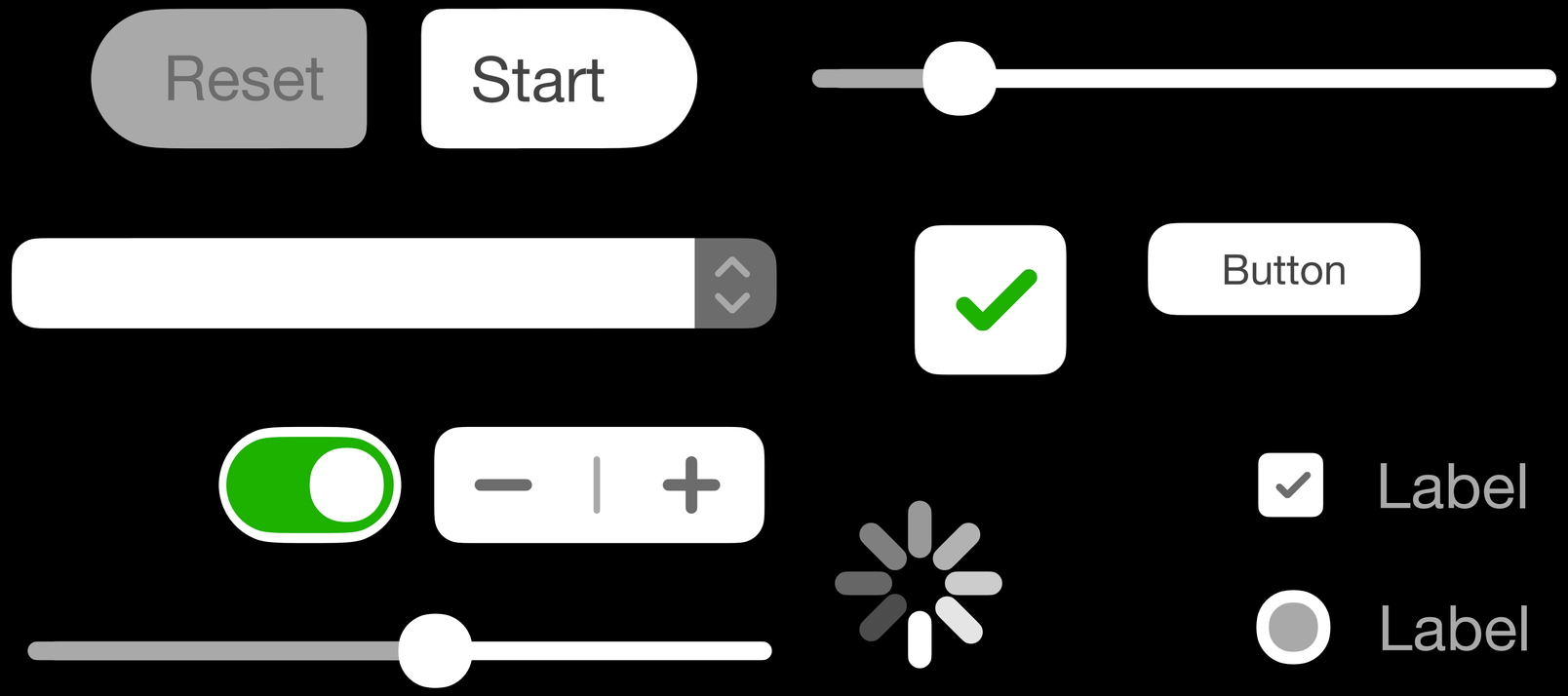
浙江大學
ZHEJIANG UNIVERSITY

控件选择的基本原则

如何完善你的App

Tips for perfecting your app

浙江大学 ZHEJIANG UNIVERSITY



如何完善你的App

Tips for perfecting your app

浙江大學
ZHEJIANG UNIVERSITY

用户能同时选中多个选项吗?

如何完善你的App

Tips for perfecting your app

如何完善你的App

Tips for perfecting your app

浙江大学 ZHEJIANG UNIVERSITY

用户能同时选中多个选项吗?

不能
(只有一个选项)

能
(有多个选项)

选项文案短且一致吗?

否

- [x] Taste good and like it
- [x] Will recommend to friends

CheckBox

是

Good Clean Cheap

Toggle Button

如何完善你的App

Tips for perfecting your app

浙江大学 ZHEJIANG UNIVERSITY

用户能同时选中多个选项吗?

不能
（只有一个选项）

能
（有多个选项）

选项文案短且一致吗?

否

是

Taste good and like it
Will recommend to friends

CheckBox

Good | Clean | Cheap

Toggle Button

选项是用来改变信息
还是筛选?

改变

筛选

Science | Art

Tabs

如何完善你的App

Tips for perfecting your app

浙江大学 ZHEJIANG UNIVERSITY

用户能同时选中多个选项吗?

不能（只有一个选项）

选项是用来改变信息还是筛选?

选择

有多少选项

1个

需要二次确认码

否

是

Toggle

Yes

CheckBox

3个

选项文案短且一致吗?

否

是

Yes

No

单选框

大于5个

1 2 3

分段选择器

能（有多个选项）

筛选

Science Art

Tabs

选项文案短且一致吗?

否

Taste good and like it

Will recommend to friends

CheckBox

是

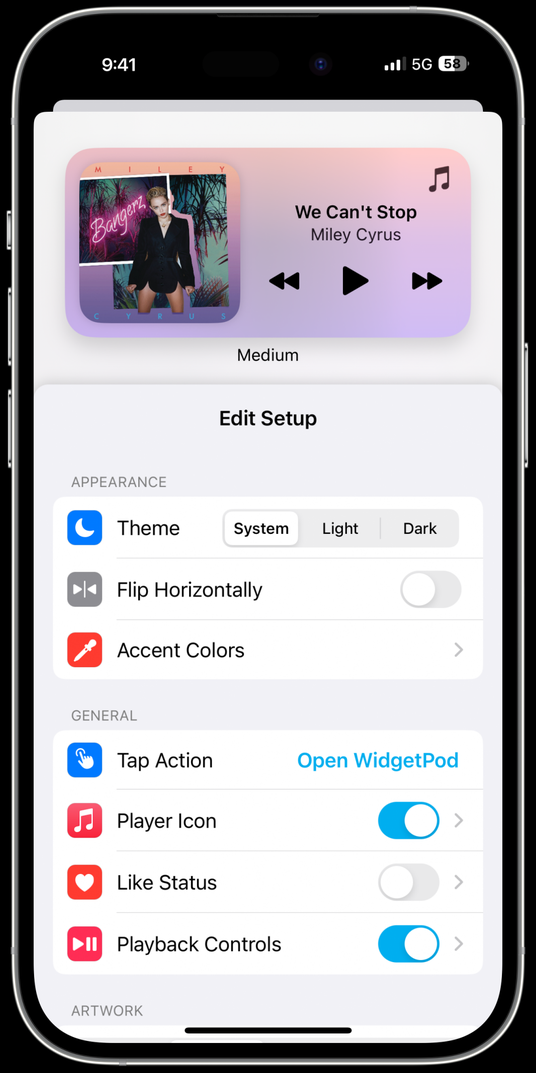
Good Clean Cheap

Toggle Button

下拉菜单

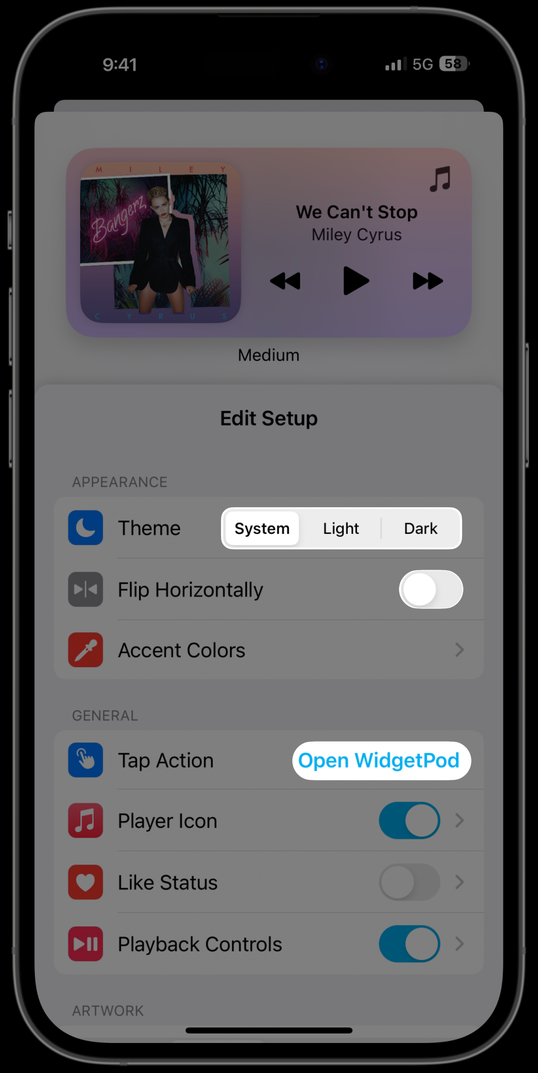
如何完善你的App

Tips for perfecting your app



如何完善你的App

Tips for perfecting your app



多个选择，但用户只能选择一个
选择都很短且一致整齐

单个选择

大于五个选择，但用户只能选择一个
采用不展开的下拉菜单

如何完善你的App

Tips for perfecting your app

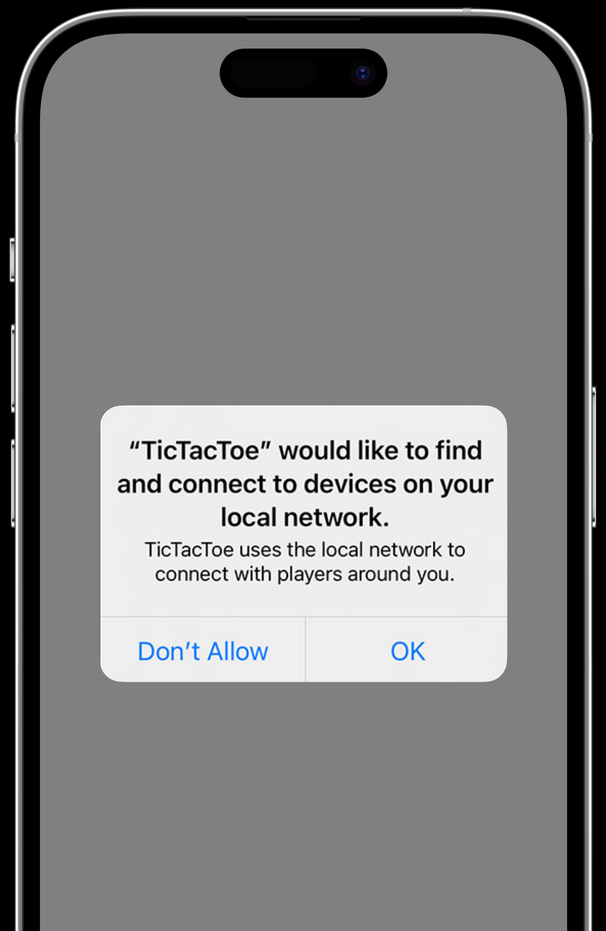
浙江大学 ZHEJIANG UNIVERSITY

在设计开始[illegible]重用户隐私

如何完善你的App

Tips for perfecting your app

浙江大学 ZHEJIANG UNIVERSITY

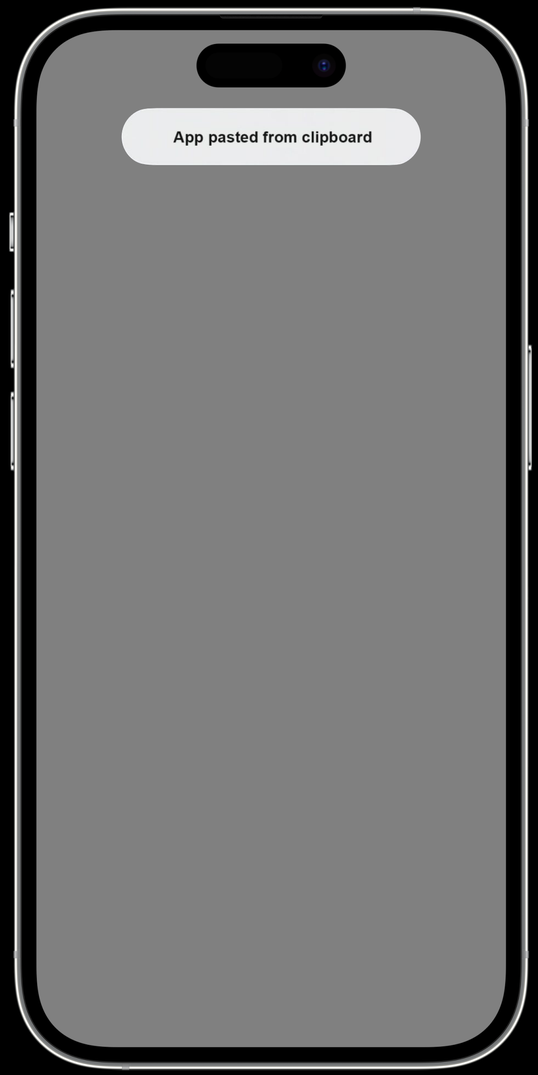


如何完善你的App

Tips for perfecting your app

浙江大学 ZHEJIANG UNIVERSITY

```
func inspectPasteboard(){
    let pasteboard = UIPasteboard.general
    if pasteboard.hasStrings {
        pastedString = pasteboard.string!
    }else if pasteboard.hasImages{
        pastedImage = pasteboard.image
    }
}
```



如何完善你的App

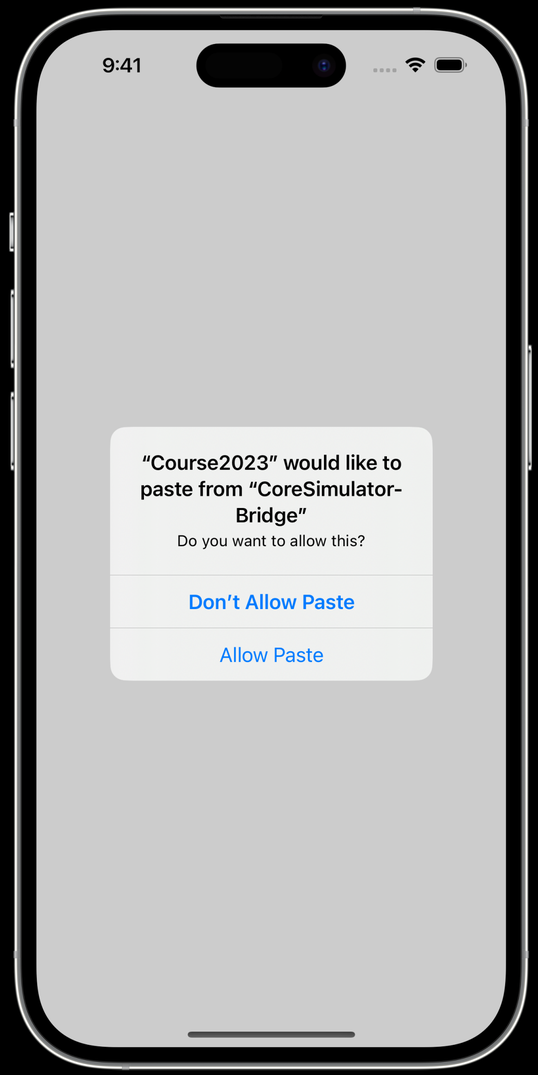
Tips for perfecting your app

```
var body: some View {
    VStack {
        Rectangle()
            .overlay{
                ZStack(alignment: .bottomLeading) {
                    Image(uiImage: pastedImage ??
                                   UIImage(named: "placeholder")!)
                        .resizable()
                    Text(pastedString)
                        .padding()
                }
            }
            .frame(width: 320, height: 240) // 4:3
        HStack{
            pasteTitleButton
            pastedImageButton
        }
    }
}
```

如何完善你的App

Tips for perfecting your app

```
var body: some View {
    VStack {
        Rectangle()
            .overlay{
                ZStack(alignment: .bottomLeading) {
                    Image(uiImage: pastedImage ??
                                        UIImage(named: "placeholder")!)
                        .resizable()
                    Text(pastedString)
                        .padding()
                }
            }
            .frame(width: 320, height: 240) // 4:3
        HStack{
            pasteTitleButton
            pastedImageButton
        }
    }

    .onAppear{
            inspectPasteboard()
    }

}
```



浙江大学
ZHEJIANG UNIVERSITY

如何完善你的App

深入理解SwiftUI中的交互设计

复杂动画的设计与实现

浙江大学
ZHEJIANG UNIVERSITY

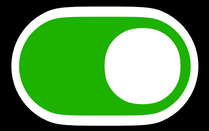
如何完善你的App

深入理解SwiftUI中的交互设计

复杂动画的设计与实现

深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI



深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

浙江大學
ZHEJIANG UNIVERSITY

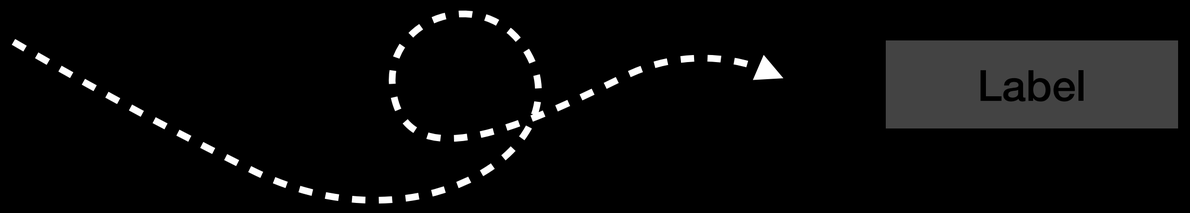
```
Button("+1") {
   print("button pressed!")
}
  .buttonStyle(.bordered)
```

深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

浙江大学 ZHEJIANG UNIVERSITY

```
struct MyButtonStyle: ButtonStyle {
    public func makeBody(configuration: MyButtonStyle.Configuration) -> some View {
        return configuration.label
    }

}
```

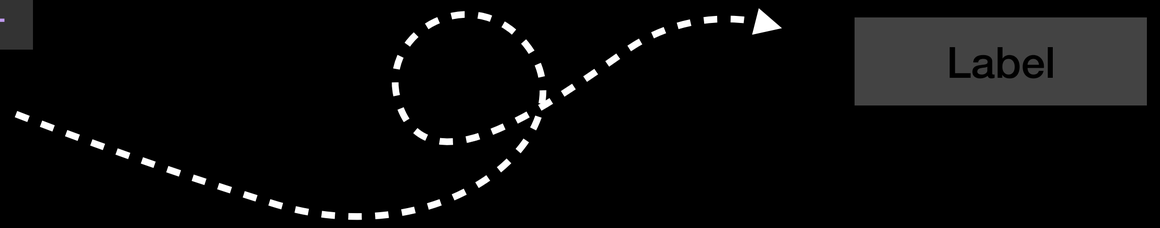


深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

浙江大学 ZHEJIANG UNIVERSITY

```
struct MyButtonStyle: ButtonStyle {

    public func makeBody(configuration: MyButtonStyle.Configuration) -> some View {
        MyButton(configuration: configuration )
    }
    struct MyButton: View {
        let configuration: MyButtonStyle.Configuration

        var body: some View {

            return configuration.label
        }
    }
}
```



深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

浙江大学 ZHEJIANG UNIVERSITY

```
struct MyButtonStyle: ButtonStyle {

    public func makeBody(configuration: MyButtonStyle.Configuration) -> some View {
        MyButton(configuration: configuration, color: color)
    }
     struct MyButton: View {
        let configuration: MyButtonStyle.Configuration

        let color: Color

        var body: some View {

            return configuration.label
                    .foregroundColor(.white)
                    .padding(15)
                    .background(RoundedRectangle(cornerRadius: 5)
                                .fill(configuration.isPressed ? .green :color))
                    .shadow(color: .black, radius: 3)
                    .opacity(configuration.isPressed ? 0.5 : 1.0)
        }

    }
}
```

Label

深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

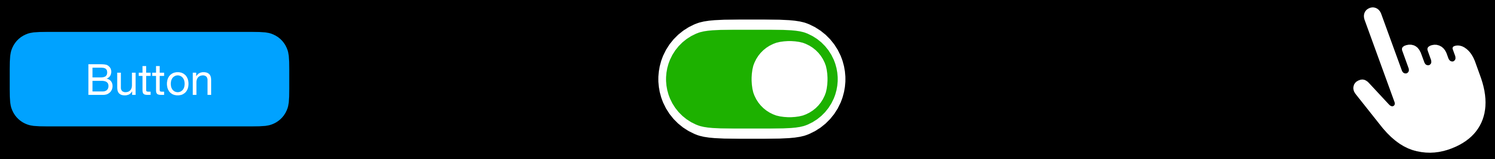
浙江大學
ZHEJIANG UNIVERSITY

```
Button("+1") {
   print("button pressed!")
}
  .buttonStyle(MyButtonStyle(color: .blue))
```

深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

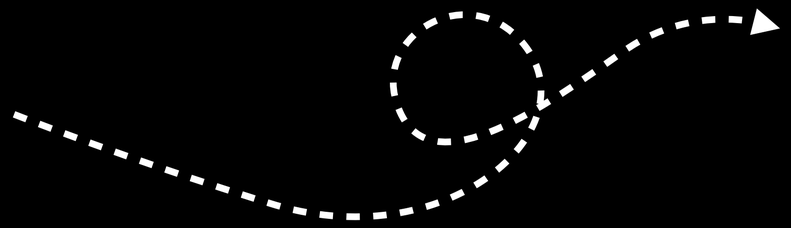
浙江大学 ZHEJIANG UNIVERSITY



深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI


```
struct SimpleToggleStyle: ToggleStyle {
    func makeBody(configuration: Configuration) -> some View {
        configuration.label
    }
}
```

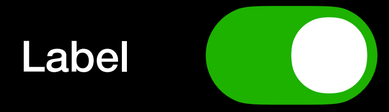


深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

浙江大学 ZHEJIANG UNIVERSITY

```
struct SimpleToggleStyle: ToggleStyle {
    func makeBody(configuration: Configuration) -> some View {
        HStack {
            configuration.label
            RoundedRectangle(cornerRadius: 20, style: .continuous)
                .fill(configuration.isOn ? .green : .black)
                .frame(width: 51, height: 31)
                .overlay{
                    Circle()
                        .foregroundColor(.white)
                        .padding(.all, 3)
                }
        }
    }
}
```

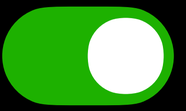


深入理解SwiftUI中的交互设计

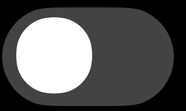
Go deep into the interaction design in SwiftUI


```
struct SimpleToggleStyle: ToggleStyle {
    func makeBody(configuration: Configuration) -> some View {
        HStack {
            configuration.label
            RoundedRectangle(cornerRadius: 20, style: .continuous)
                .fill(configuration.isOn ? .green : .black)
                .frame(width: 51, height: 31)
                .overlay{
                    Circle()
                        .foregroundColor(.white)
                        .padding(.all, 3)
                        .offset(x: configuration.isOn ? 11 : -11, y: 0)
                }
                .onTapGesture {
                    withAnimation(Animation.linear(duration: 0.1)){
                        configuration.isOn.toggle()
                    }
                }
        }
    }
}
```

Label

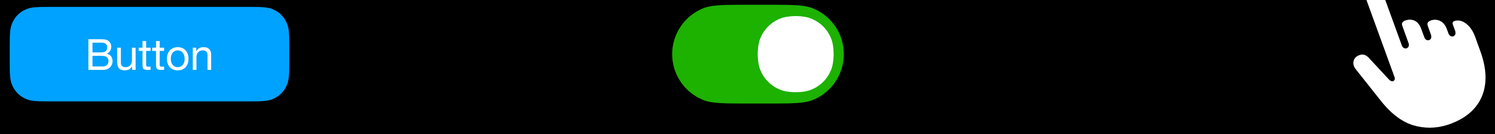


Label



深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI



深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

TapGesture

LongPressGesture

DragGesture

CustomGesture

深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

浙江大學
ZHEJIANG UNIVERSITY

TapGesture

LongPressGesture

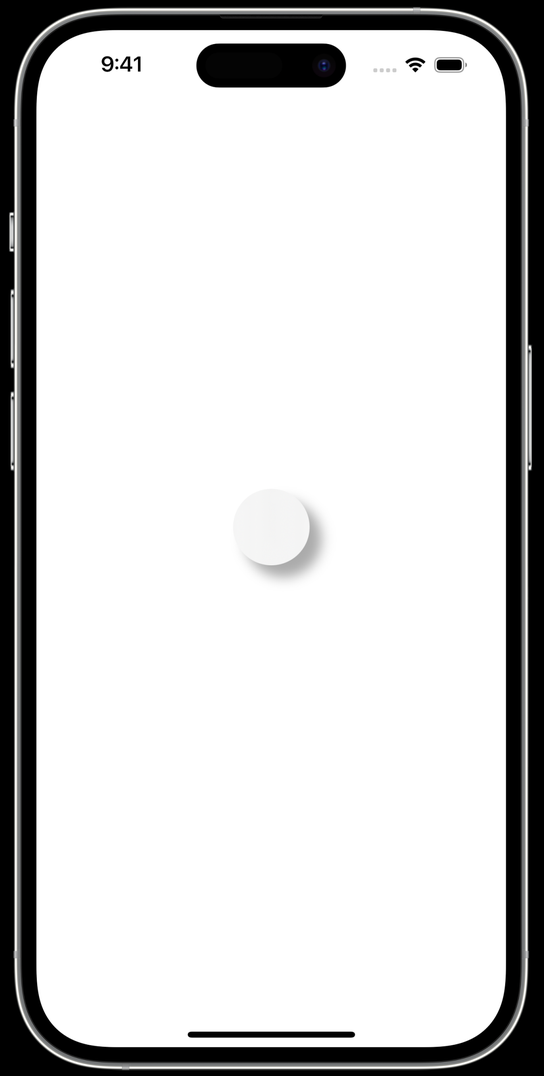
DragGesture

CustomGesture

深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI


```
struct GestureView: View {

    var body: some View {

        CustomCircle()

    }
}
```



深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

浙江大学 ZHEJIANG UNIVERSITY

```
struct GestureView: View {

    @State private var isActivated = false
    @State private var isLongPressing = false

    var body: some View {

        VStack {
            CustomCircle()
                .scaleEffect(isActivated ? 1.5 : 1)
                .onLongPressGesture(minimumDuration: 1,
                                    pressing: { pressing in isLongPressing = pressing },
                                    perform: { withAnimation {isActivated.toggle()} })

            Text("Hold on...")
                .foregroundColor(isLongPressing ? .green : .clear)
                .padding(.top)
        }
    }
}
```

深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

浙江大学 ZHEJIANG UNIVERSITY

```
struct GestureView: View {
    @GestureState private var isDetectingLongPress = false

    var longPressGesture: some Gesture {
        LongPressGesture(minimumDuration: 2)
            .updating($isDetectingLongPress){currentState, gestureState, pressing
                transaction in
                gestureState = currentState
                transaction.animation = Animation.easeIn(duration: 2.0)
            }
                                                perform
            .onEnded { value in
                withAnimation {
                    isActivated = value
                }
            }
    }
```

深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI


```
struct GestureView: View {
    @State private var offset = CGSize.zero
    @State private var location: CGPoint = CGPoint(x: 200, y: 100)
    var longPressGesture: some Gesture { }
    var dragGesture: some Gesture {
        DragGesture()
            .onChanged { value in                    dragging
                offset = value.translation

            }
            .onEnded { value in                      perform
                withAnimation {
                    offset = .zero
                    location = value.location
                }
            }
    }
}
```

深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI


```
struct GestureView: View {

    var longPressGesture: some Gesture {  }

    var dragGesture: some Gesture { }

    var combinedGesture: some Gesture {
            longPressGesture.sequenced(before: dragGesture)
    }

    var body: some View {

     Circle()
         .fill(.ultraThinMaterial)
         .frame(width: 64, height: 64)
         .scaleEffect(isDragging ? 1.5 : 1)
         .offset(offset)
         .position(location)
         .shadow(radius: 5, x: 10, y:10)
         .gesture(combinedGesture)

    }
```

深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

结合视觉、触觉、听觉打造完美App

深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

为了产生有效的反馈，必须让用户清楚的感知到是什么导致的反馈

For feedback to be useful, it must be obvious what caused it.

触觉反馈必须与视觉、听觉设计达成和谐统一

Haptic design should feels the way it looks and the way it sounds

只有在正真需要触觉反馈时才使用它

Add audio and haptics that provide clear value to your app experience

深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

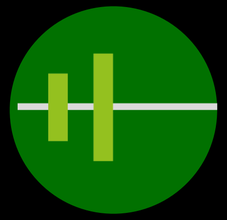
浙江大學
ZHEJIANG UNIVERSITY

触觉

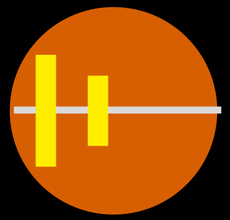
深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

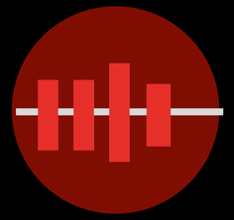
浙江大学 ZHEJIANG UNIVERSITY



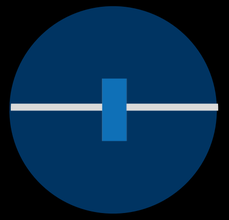
Success



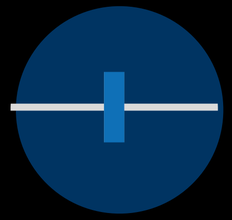
Warning



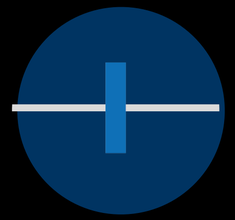
Error



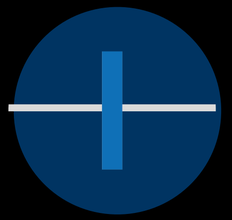
Light



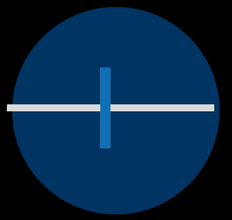
Medium



Heavy



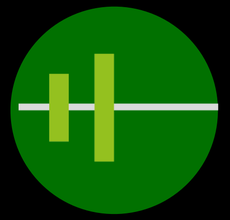
Rigid



Soft

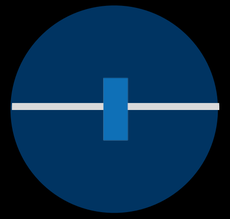
深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI



Success

```
let NFGenerator = UIImpactFeedbackGenerator()

NFGenerator.notificationOccurred(.success)
```



Light

```
let generator = UINotificationFeedbackGenerator(style: .light)
generator.impactOccurred()
```

深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

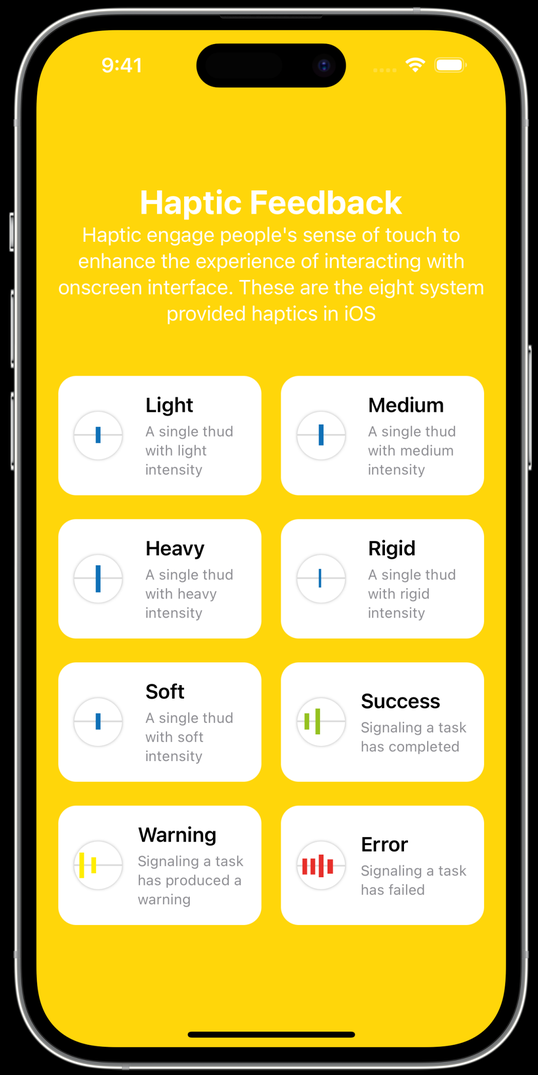
浙江大学 ZHEJIANG UNIVERSITY

UIImpactFeedbackGenerator

Success　　Warning　　Error

UINotificationFeedbackGenerator

Light　　Medium　　Heavy　　Rigid　　Soft



深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

如何实现复杂的 Haptic 设计?

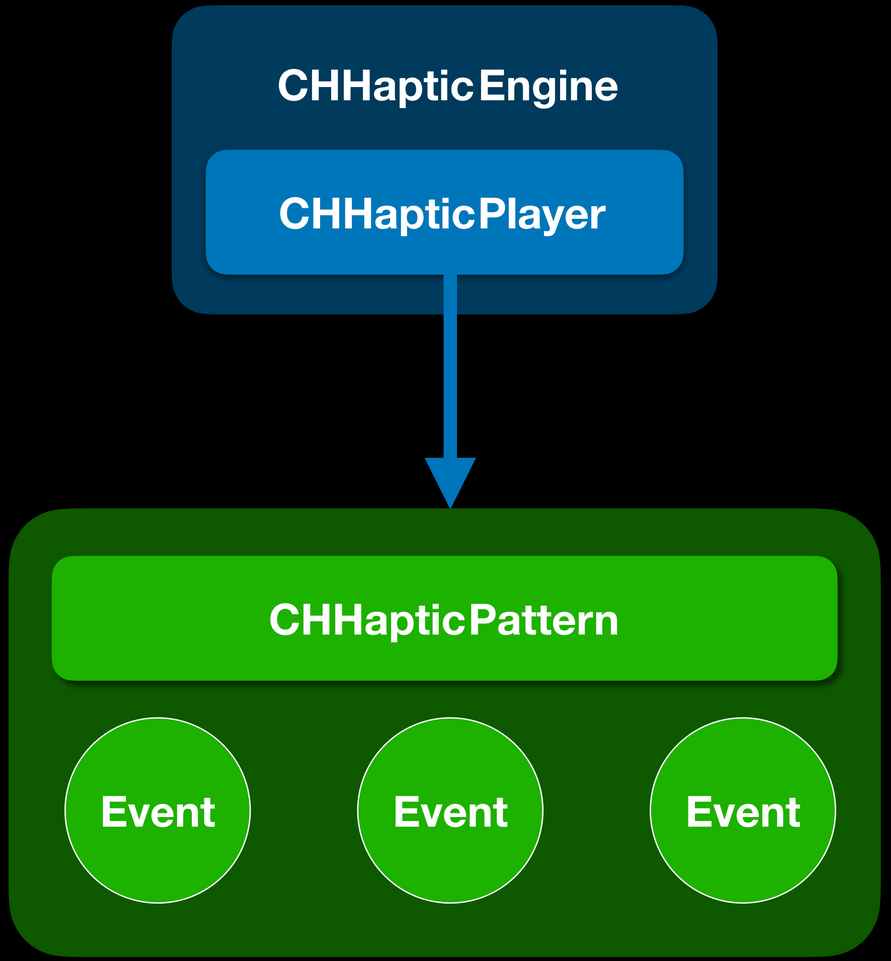
深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

使用 Core Haptics！

深入理解SwiftUI中的交互设计

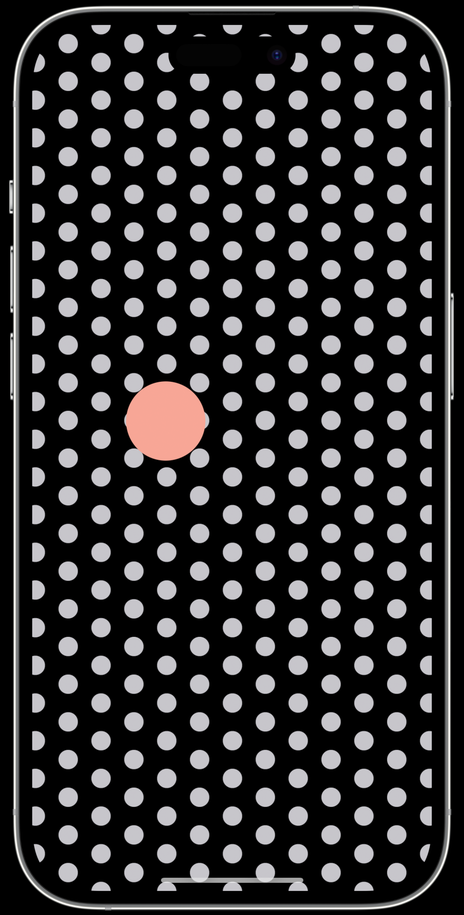
Go deep into the interaction design in SwiftUI



深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

浙江大学 ZHEJIANG UNIVERSITY



深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

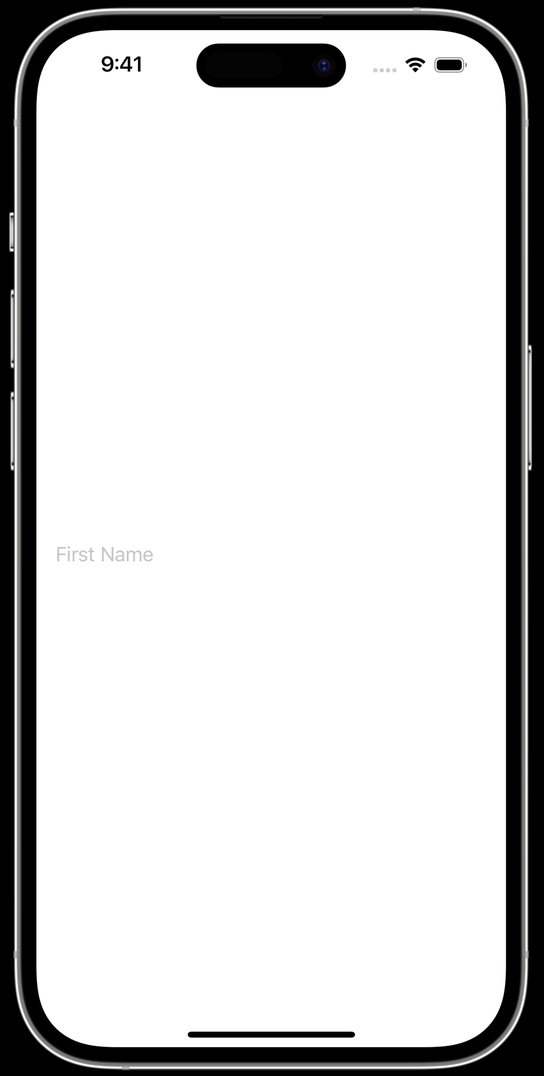
键盘的布局与选择?

深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

浙江大学 ZHEJIANG UNIVERSITY

```
struct keyboardView: View {
    @State private var firstname: String = ""

    var body: some View {

        TextField("First Name",
                  text: $firstname)

            .padding()
     }
}
```

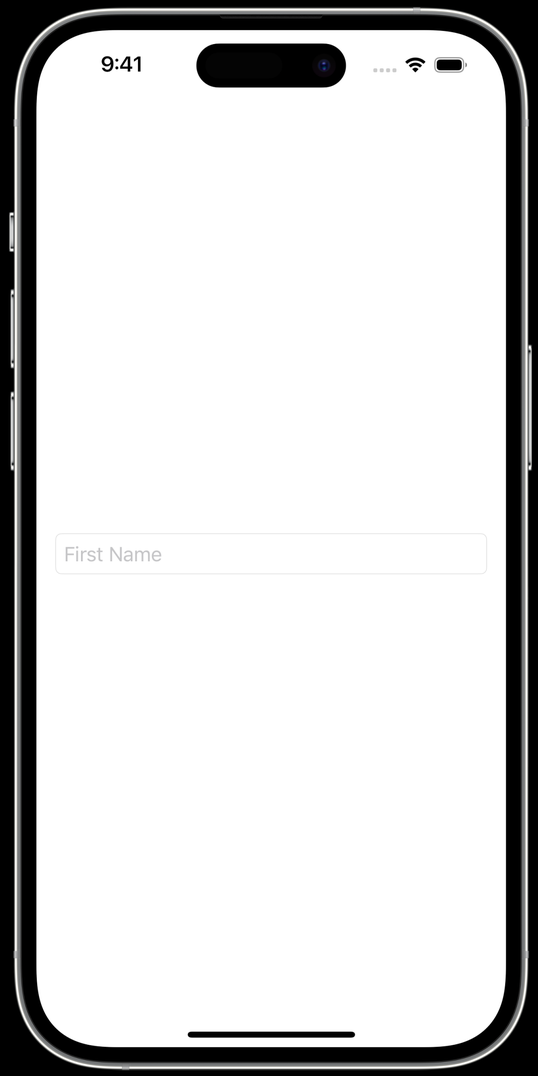


深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

浙江大学 ZHEJIANG UNIVERSITY

```
struct keyboardView: View {
    @State private var firstname: String = ""

    var body: some View {
       TextField("First Name",
                 text: $firstname )
           .padding()
           .textFieldStyle(.roundedBorder)
     }
}
```

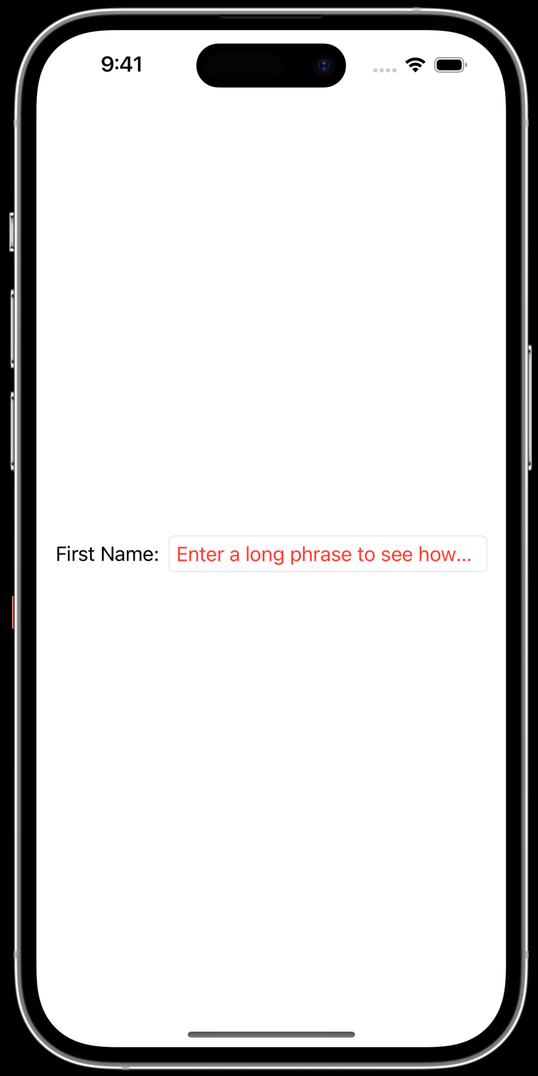


深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

浙江大学 ZHEJIANG UNIVERSITY

```
struct keyboardView: View {
    @State private var firstname: String = ""

    var body: some View {
        HStack{
                Text("First Name:")
                TextField("First Name",
                          text: $firstname ,
                          prompt: Text("Enter a long phrase to see h
                                  .foregroundColor(.red),
                          axis: .vertical)
                    .textFieldStyle(.roundedBorder)
        }
        .padding()
    }
}
```

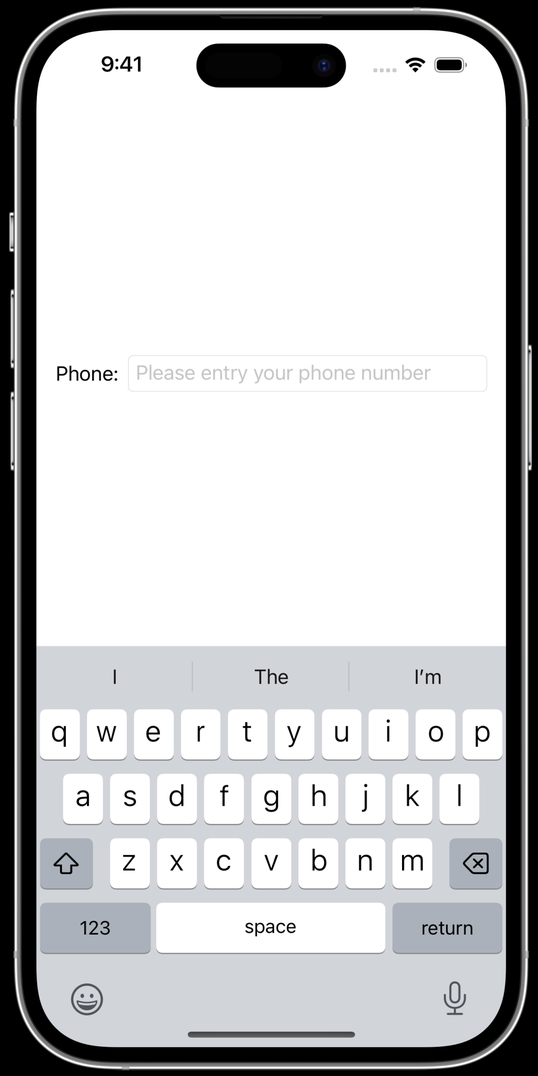


深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

浙江大学 ZHEJIANG UNIVERSITY

```
struct keyboardView: View {
    @State private var phone: String = ""

    var body: some View {
        HStack{
                Text("Phone:")
                TextField("Please entry your phone number",
                          text: $phone )

                   .textFieldStyle(.roundedBorder)
                   .keyboardType(.default)
        }
        .padding()
    }
}
```



深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

浙江大学 ZHEJIANG UNIVERSITY

```
struct keyboardView: View {
    @State private var phone: String = ""

    var body: some View {
        HStack{
                Text("Phone:")
                TextField("Please entry your phone number",
                          text: $phone )
                    .textFieldStyle(.roundedBorder)
                    .keyboardType(.phonePad)
        }
        .padding()
    }
}
```

深入理解SwiftUI中的交互设计

Go deep into the interaction design in SwiftUI

浙江大學
ZHEJIANG UNIVERSITY

```
.phonePad
.alphabet
.asciiCapable
.asciiCapableNumberPad
.numberPad
.URL
.default
.decimalPad
.emailAddress
.numbersAndPunctuation
.twitter
.webSearch
```

浙江大学
ZHEJIANG UNIVERSITY

如何完善你的App

深入理解SwiftUI中的交互设计

复杂动画的设计与实现

浙江大学
ZHEJIANG UNIVERSITY

如何完善你的App

深入理解SwiftUI中的交互设计

复杂动画的设计与实现

复杂动画的设计与实现

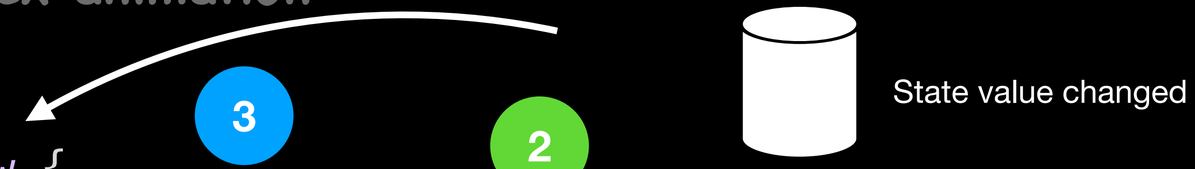
Design and code complex animation

浙江大学
ZHEJIANG UNIVERSITY

找到变化的源头，对其变化过程进行优化

复杂动画的设计与实现

Design and code complex animation




```
struct StateView: View {
    @State private var name = "?"
    private let names = ["Mark", "Chris", "Scott", "Sean", "Paul"]

    var body: some View {
        VStack(spacing: 20) {
            Text("Hi, my name is: \(name)")
                .font(.title)

            Button("Random Name") {
                self.name = self.names[Int.random(in: 0...4)]
            }
            .buttonStyle(.bordered)
        }
    }
}
```

复杂动画的设计与实现

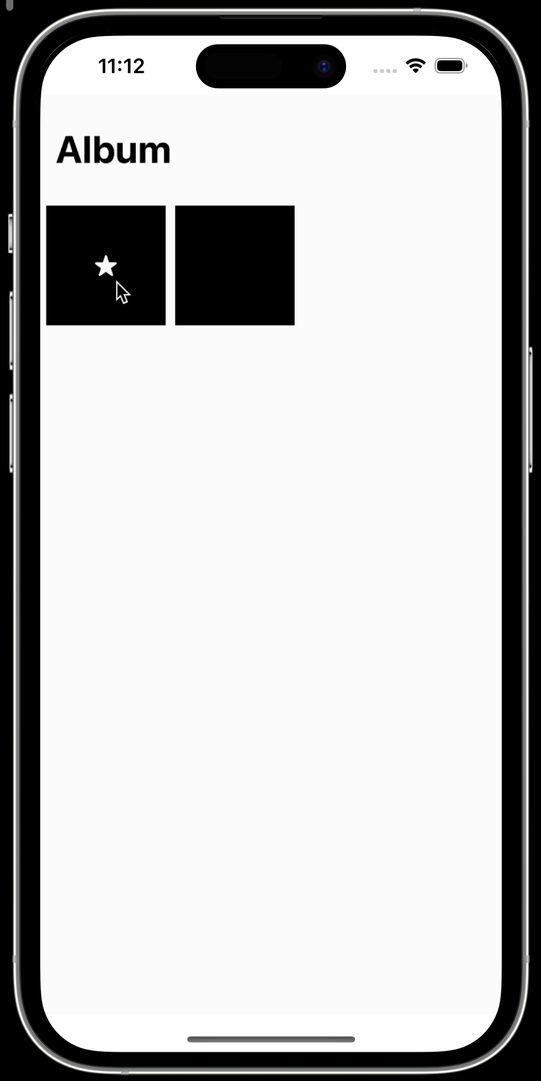
Design and code complex animation

浙江大学
ZHEJIANG UNIVERSITY

```
struct StateView: View {
    @State private var name = "?"
    private let names = ["Mark", "Chris", "Scott", "Sean", "Paul"]

    var body: some View {
        VStack(spacing: 20) {
            Text("Hi, my name is: \(name)")
                .font(.title)

            Button("Random Name") {
                withAnimation(.linear){
                    self.name = self.names[Int.random(in: 0...4)]
                }
            }
            .buttonStyle(.bordered)
        }
    }
}
```

复杂动画的设计与实现

Design and code complex animation

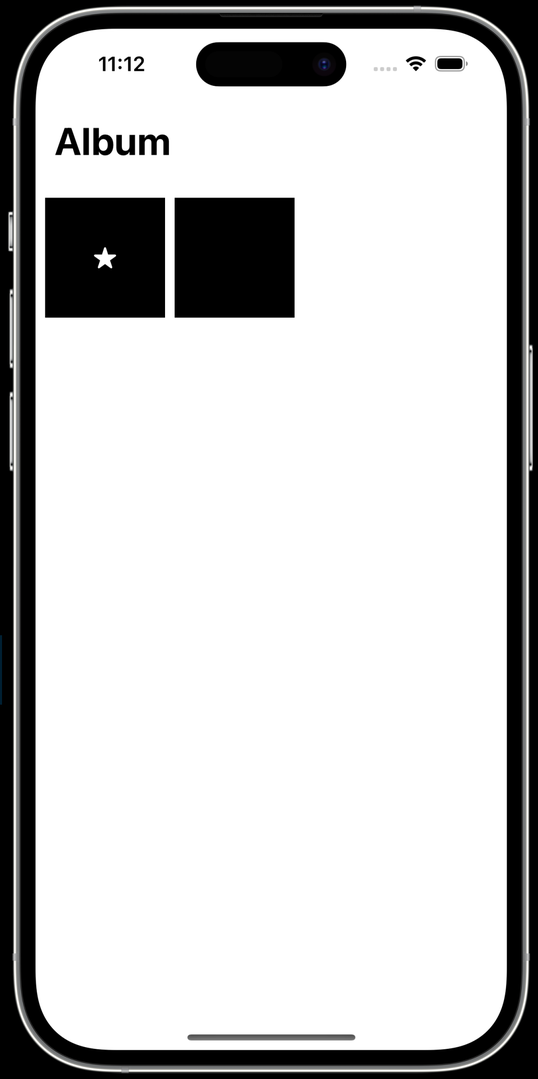


复杂动画的设计与实现

Design and code complex animation

浙江大学
ZHEJIANG UNIVERSITY

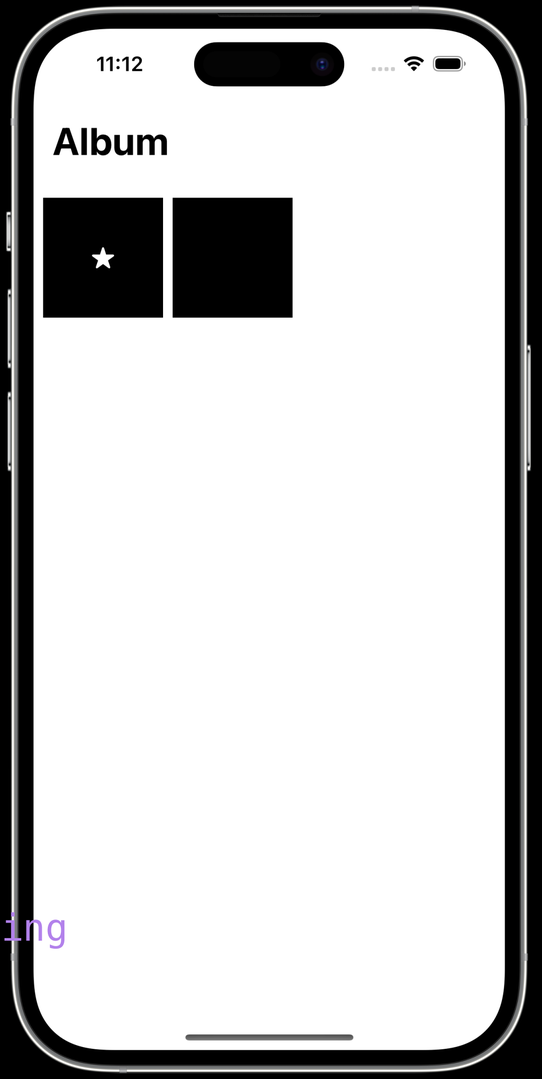
```
@Namespace var namespace


var Photo: some View{
    Image(systemName: "star.fill")
        .foregroundColor(.white)
        .matchedGeometryEffect(id: "title", in: namespace)
        .frame(width: 100, height: 100)
        .background(
            Rectangle()
                .matchedGeometryEffect(id: "shape", in: namespace)
        )
}
```



复杂动画的设计与实现

Design and code complex animation

```
var Photo: some View{
    Image(systemName: "star.fill")
        .foregroundColor(.white)
        .matchedGeometryEffect(id: "title", in: namespace)
        .frame(width: 100, height: 100)
        .background(
            Rectangle()
                .matchedGeometryEffect(id: "shape", in: namespace)
        )
}

var FullScreenPhoto: some View{
    ZStack {
        Rectangle()
            .matchedGeometryEffect(id: "shape", in: namespace)
            .frame(maxWidth: .infinity, maxHeight: 400)

        Image(systemName: "star.fill")
            .foregroundColor(.white)
            .matchedGeometryEffect(id: "title", in: namespace)
            .frame(maxWidth: .infinity, maxHeight: 350, alignment: .topLeading)
            .padding()

    }
}
```



复杂动画的设计与实现

Design and code complex animation

浙江大学 ZHEJIANG UNIVERSITY

```
var body: some View {
    ZStack {
        if !showFullScreenPhoto{
            Photo
                .onTapGesture {
                    withAnimation(.spring(response: 0.5, dampingFraction: 0.7)) {
                        showFullScreenPhoto.toggle()
                    }
                }
        }else if showFullScreenPhoto{
            FullScreenPhoto
        }
    }
    .onTapGesture {
        withAnimation(.spring(response: 0.5, dampingFraction: 0.7)) {
            showFullScreenPhoto = false
        }
    }
}
```

复杂动画的设计与实现

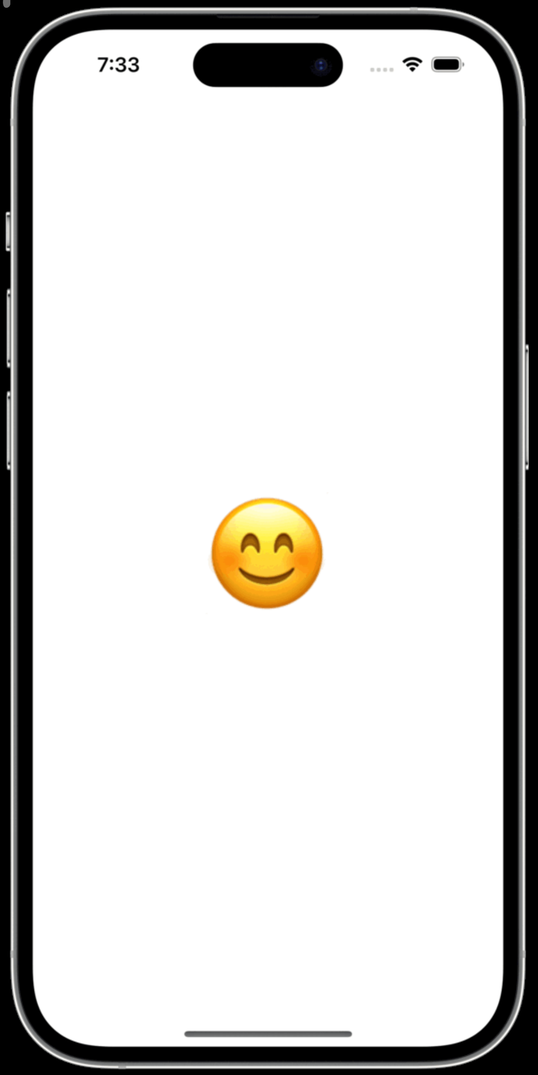
Design and code complex animation

浙江大学 ZHEJIANG UNIVERSITY

“非显性”变化的源头导致的变化该怎么添加动画?

复杂动画的设计与实现

Design and code complex animation



复杂动画的设计与实现

Design and code complex animation


```
struct AnimationView: View {
    static let emoji = ["☺️", "😊", "😄", "😁", "😆", "😂", "🤣", "🤪"]

    var body: some View {
        TimelineView(.periodic(from: .now, by: 0.2)) { _ in

            let randomEmoji = AnimationView.emoji.randomElement() ?? ""

            Text(randomEmoji)
                .font(.largeTitle)
                .scaleEffect(3.0)

        }

    }

}
```

复杂动画的设计与实现

Design and code complex animation

浙江大学 ZHEJIANG UNIVERSITY

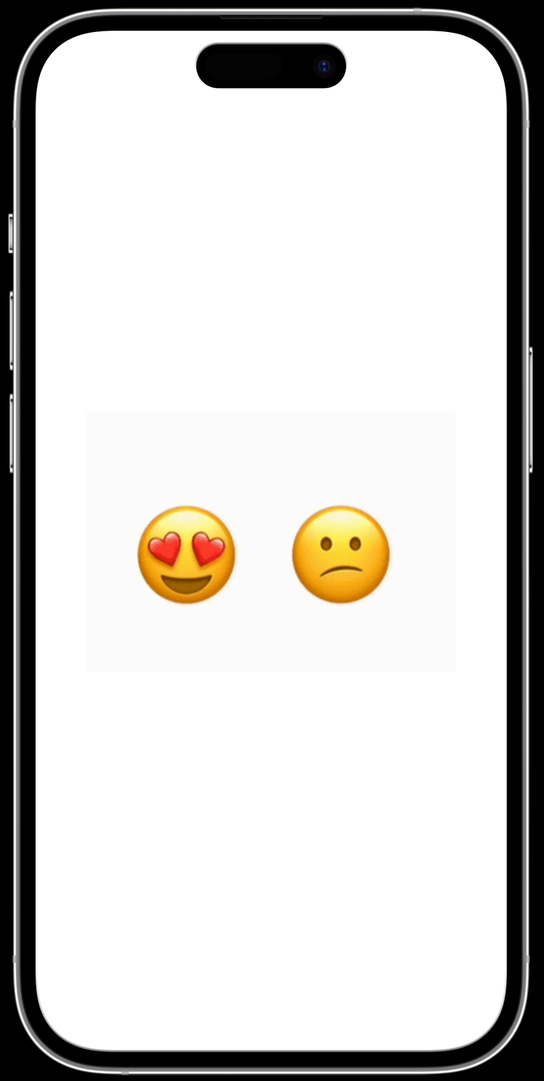
```
struct AnimationView: View {
    static let emoji = ["☺️", "😊", "😄", "😁", "😆", "😂", "🤣", "🤪"]

    var body: some View {
        TimelineView(.periodic(from: .now, by: 0.2)) { _ in
            HStack(spacing: 100) {
                let randomEmoji = AnimationView.emoji.randomElement() ?? ""

                Text(randomEmoji)
                    .font(.largeTitle)
                    .scaleEffect(3.0)
                subView
            }
        }
    }
    var subView: some View {

        let randomEmoji = AnimationView.emoji.randomElement() ?? ""

        return Text(randomEmoji)
            .font(.largeTitle)
            .scaleEffect(3.0)
    }
}
```

?

复杂动画的设计与实现

Design and code complex animation

```
struct AnimationView: View {
    static let emoji = ["☺️", "😊", "😄", "😁", "😆", "😂", "🤣", "🤪"]

    var body: some View {
        TimelineView(.periodic(from: .now, by: 0.2)) { _ in
            HStack(spacing: 100) {
                let randomEmoji = AnimationView.emoji.randomElement() ?? ""

                Text(randomEmoji)
                    .font(.largeTitle)
                    .scaleEffect(3.0)
                SubView()
            }
        }
    }
}

struct SubView: View {
    var body: some View {
        let randomEmoji = AnimationView.emoji.randomElement() ?? ""

        Text(randomEmoji)
            .font(.largeTitle)
            .scaleEffect(3.0)
    }
}
```



复杂动画的设计与实现

Design and code complex animation

浙江大学 ZHEJIANG UNIVERSITY

视图的变化需要依赖可能存在的“变化”

复杂动画的设计与实现

Design and code complex animation

```
struct AnimationView: View {
    static let emoji = ["☺️", "😊", "😄", "😄", "😆", "😂", "🤣", "🤪"]

    var body: some View {
        TimelineView(.periodic(from: .now, by: 0.2)) { _ in
            HStack(spacing: 100) {
                let randomEmoji = AnimationView.emoji.randomElement() ?? ""

                Text(randomEmoji)
                    .font(.largeTitle)
                    .scaleEffect(3.0)
                SubView(date: timeline.date)
            }
        }
    }
}
struct SubView: View {
    let date: Date
    var body: some View {
        let randomEmoji = AnimationView.emoji.randomElement() ?? ""

        Text(randomEmoji)
            .font(.largeTitle)
            .scaleEffect(3.0)
    }
}
```

复杂动画的设计与实现

Design and code complex animation

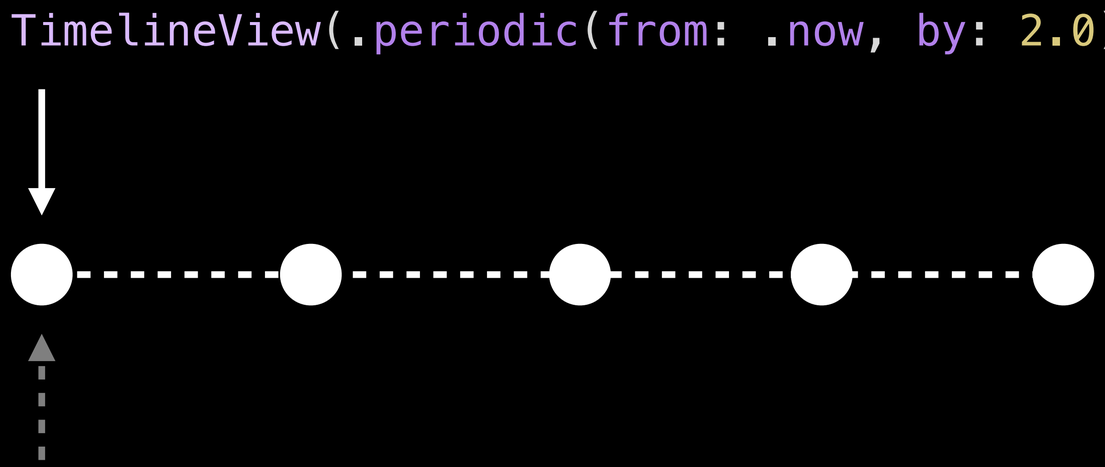
浙江大學
ZHEJIANG UNIVERSITY

```
struct TheView: View {
    @State private var flag = false

    var body: some View {

        TimelineView(.periodic(from: .now, by: 2.0)) { timeline in

            Text("Hello")
                .foregroundStyle(flag ? .red : .blue)
                .onChange(of: timeline.date) { (date: Date) in
                    flag.toggle()
                }

        }
    }
}
```

```
onChange(of: Date) action
tried to update multiple times per frame.
```

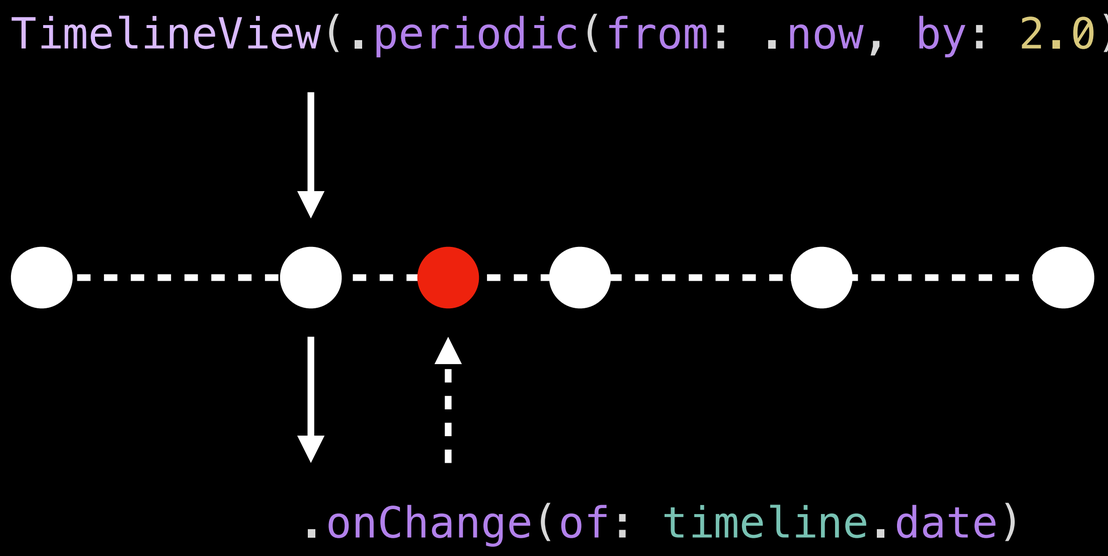
复杂动画的设计与实现

Design and code complex animation



复杂动画的设计与实现

Design and code complex animation



复杂动画的设计与实现

Design and code complex animation


```
struct TheView: View {
    var body: some View {

        TimelineView(.periodic(from: .now, by: 1.0)) { timeline in
            TheSubView(date: timeline.date)
        }

    }
}
struct TheSubView: View {
    @State private var flag = false
    let date: Date

    var body: some View {
        Text("Hello")
            .foregroundStyle(flag ? .red : .blue)
            .onChange(of: date) { (date: Date) in
                flag.toggle()
            }
    }
}
```

复杂动画的设计与实现

Design and code complex animation

浙江大学 ZHEJIANG UNIVERSITY

复杂动画的设计与实现

Design and code complex animati

浙江大学 ZHEJIANG UNIVERSITY

复杂动画的设计与实现

Design and code complex animation


```
struct CloudsView: View {
    var cloudGroup: CloudGroup

    var body: some View {
        TimelineView(.animation) { timeline in
            Canvas { context, size in
                cloudGroup.update(date: timeline.date)

                 //...

                for cloud in cloudGroup.clouds {
                    // Draw those clouds
                }
            }
    }
}
```

浙江大學
ZHEJIANG UNIVERSITY

Any Questions ?