Miguel Ruiz

iOS Developer in Cornershop


@migueloruiz


@hola_miguelo

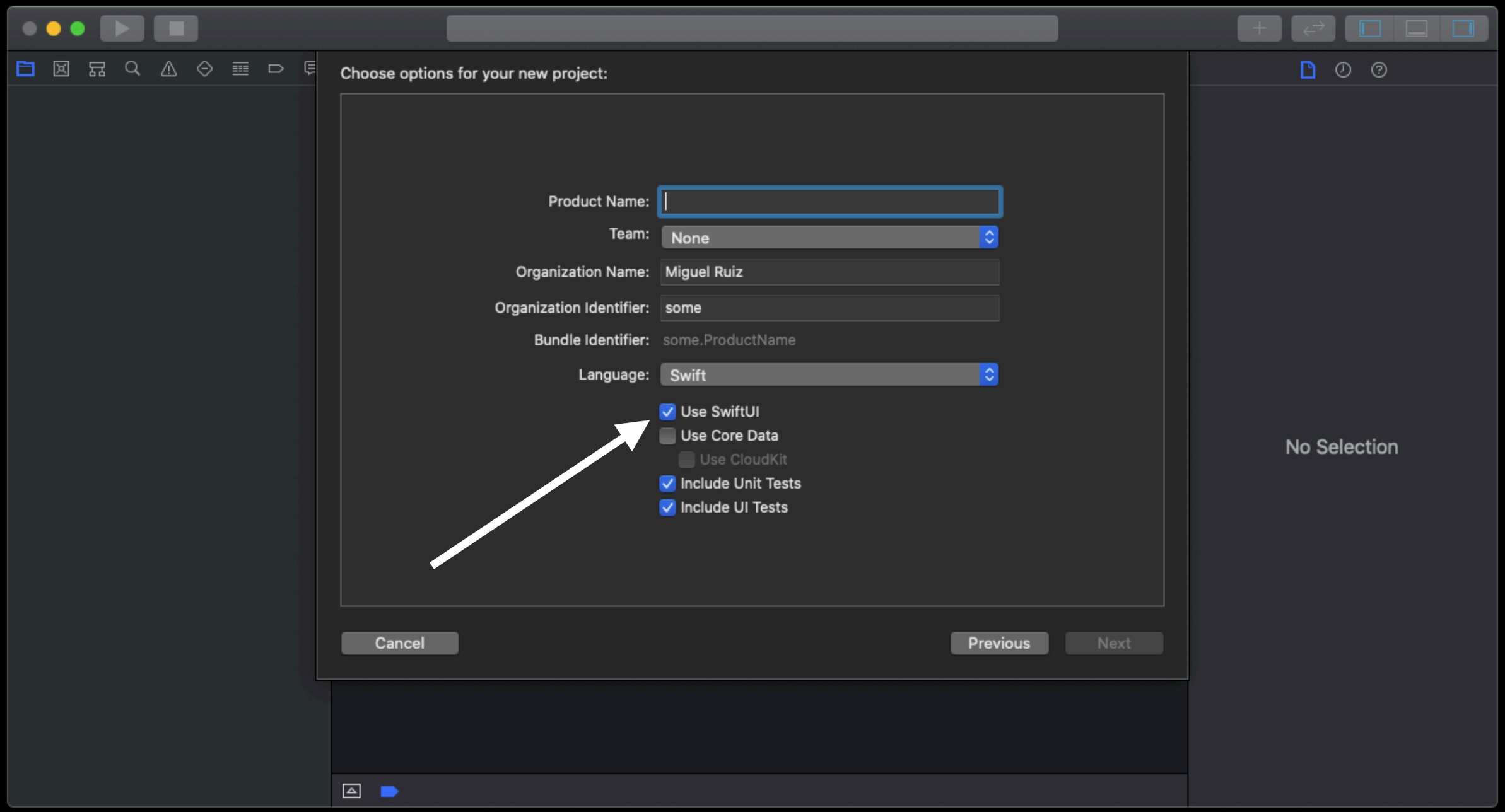
Explorando SwiftUI

Disclaimer

- Xcode 11 Beta 5
- Mac OS Catalina Beta 10.15


7:45
Cards
Mi Saldo
bip!
876987696
Saldo
2309
Actualizado
Hace 30 m...
Actualizar
Recarga

Activar SwiftUI

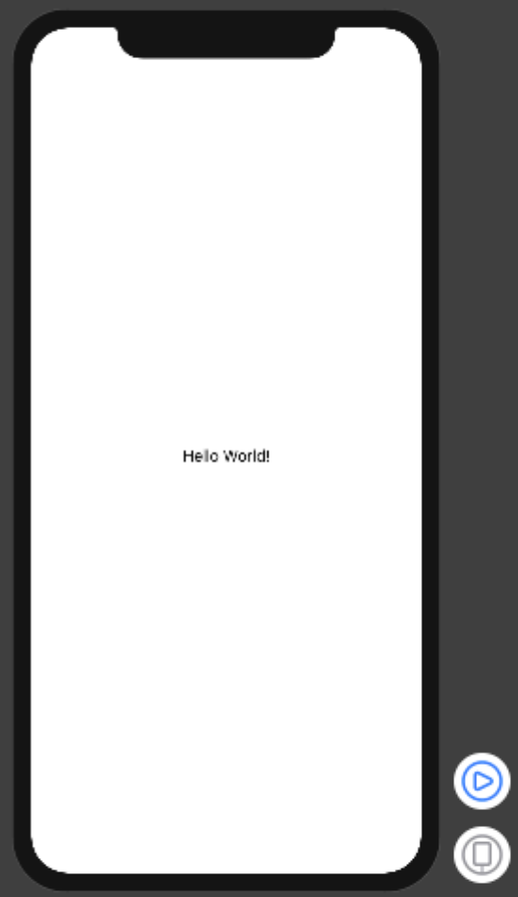


Cards 〉 iPhone XR

Cards | Preview Cards: **Succeeded** | Today at 23:25

Cards 〉 Cards 〉 Card.swift 〉 No Selection

```swift

import SwiftUI

struct CardView: View {
    var body: some View {
       Text("Hello World!")
    }
}

#if DEBUG
struct CardView_Previews : PreviewProvider {
    static var previews: some View {
        CardView()
    }
}
#endif

```



Preview

50%

OpaqueTypes

```
3
4  struct CardView: View {
5      var body: some View {
6          Text("Hello World!")
7      }
8  }
9
```

Swift evolution proposal - 0244

OpaqueTypes Swift 5.1

View

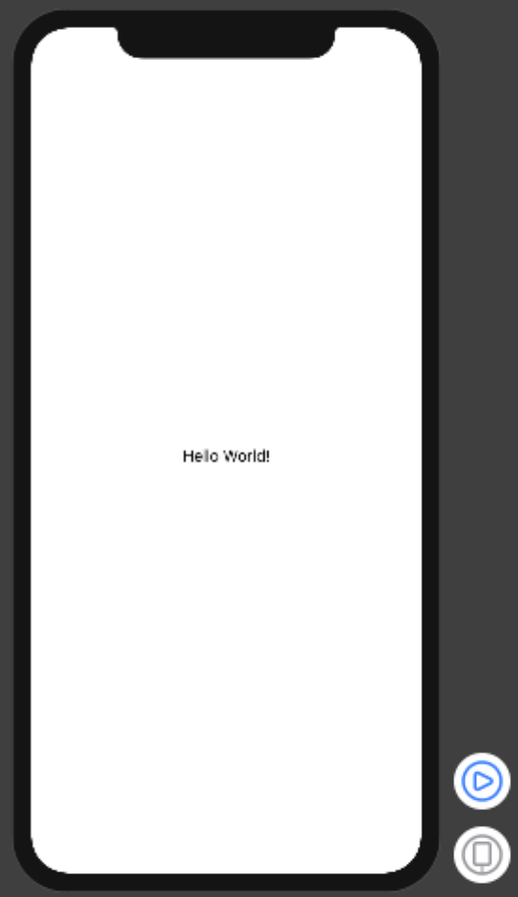
```
9338  /// A piece of user interface.
9339  ///
9340  /// You create custom views by declaring types that conform to the `View`
9341  /// protocol. Implement the required `body` property to provide the content
9342  /// and behavior for your custom view.
9343  @available(iOS 13.0, OSX 10.15, tvOS 13.0, watchOS 6.0, *)
9344  public protocol View {
9345
9346      /// The type of view representing the body of this view.
9347      ///
9348      /// When you create a custom view, Swift infers this type from your
9349      /// implementation of the required `body` property.
9350      associatedtype Body : View
9351
9352      /// Declares the content and behavior of this view.
9353      var body: Self.Body { get }
9354  }
```

Cards 〉 iPhone XR

Cards | Preview Cards: **Succeeded** | Today at 23:25

Cards 〉 Cards 〉 Card.swift 〉 No Selection

```swift

import SwiftUI

struct CardView: View {
    var body: some View {
       Text("Hello World!")
    }
}

#if DEBUG
struct CardView_Previews : PreviewProvider {
    static var previews: some View {
        CardView()
    }
}
#endif

```



Preview

50%

Cards 〉 iPhone XR

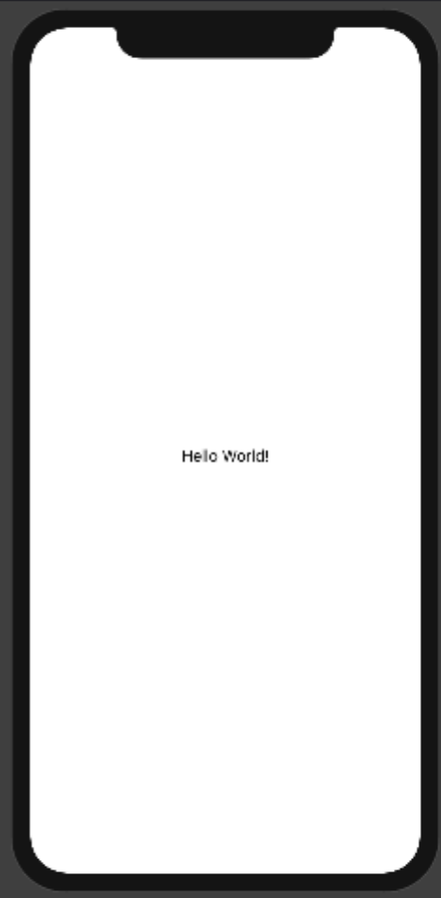
Cards | Preview Cards: **Failed** | Today at 23:02

Cards 〉 Cards 〉 CardIDLabel.swift 〉 No Selection

```
1
2  import SwiftUI
3
4  struct CardIDLabel: View {
5      let value: String
6      var body: some View {
7          Text(self.value)
8      }
9  }
10
11 #if DEBUG
12 struct CardIDLabel_Previews: PreviewProvider {
13     static var previews: some View {
14         CardIDLabel()        Missing argument for parameter 'value' in call
15     }
16 }
17 #endif
18
```

Failed to build the scheme "Cards"

Try Again | Diagnostics



Preview

50%

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:03

Cards 〉 Cards 〉 CardIDLabel.swift 〉 No Selection

```swift
import SwiftUI

struct CardIDLabel: View {
    let value: String
    var body: some View {
        Text(self.value)
    }
}

#if DEBUG
struct CardIDLabel_Previews: PreviewProvider {
    static var previews: some View {
        CardIDLabel(value: "34567453346")
    }
}
#endif
```

Preview

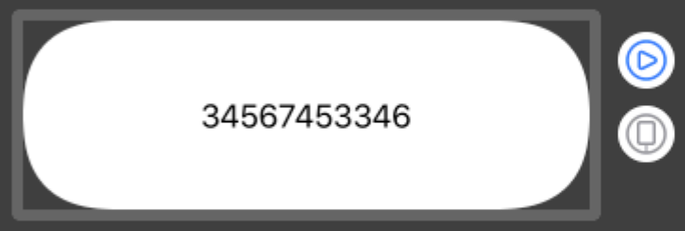
50%

Previews

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:03

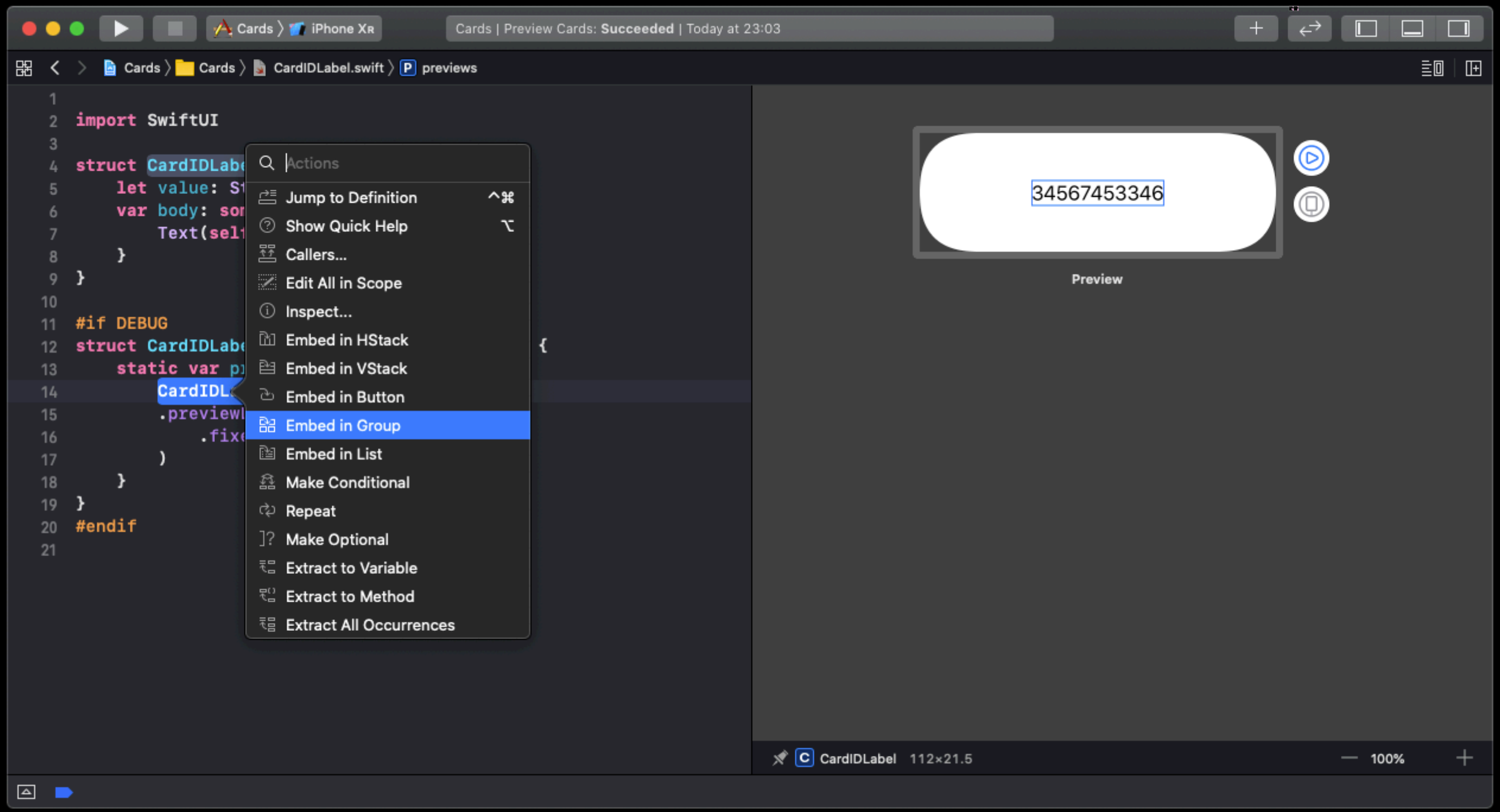
Cards 〉 Cards 〉 CardIDLabel.swift 〉 No Selection

```swift

import SwiftUI

struct CardIDLabel: View {
    let value: String
    var body: some View {
        Text(self.value)
    }
}

#if DEBUG
struct CardIDLabel_Previews: PreviewProvider {
    static var previews: some View {
        CardIDLabel(value: "34567453346")
        .previewLayout(
            .fixed(width: 300, height: 100)
        )
    }
}
#endif

```



Preview

100%


Cards 〉 iPhone XR
Cards | Preview Cards: Succeeded | Today at 23:03
Cards 〉 Cards 〉 CardIDLabel.swift 〉 previews
import SwiftUI
struct CardIDLabe
let value: St
var body: sor
Text(selt
}
}
#if DEBUG
struct CardIDLabe
static var pr
CardIDL
.previewL
.fixe
)
}
}
#endif
Actions
Jump to Definition ^⌘
Show Quick Help ⌥
Callers...
Edit All in Scope
Inspect...
Embed in HStack
Embed in VStack
Embed in Button
Embed in Group
Embed in List
Make Conditional
Repeat
Make Optional
Extract to Variable
Extract to Method
Extract All Occurrences
34567453346
Preview
CardIDLabel 112×21.5
100%

⌘ + Click

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:03

Cards 〉 Cards 〉 CardIDLabel.swift 〉 No Selection

```swift

import SwiftUI

struct CardIDLabel: View {
    let value: String
    var body: some View {
        Text(self.value)
    }
}

#if DEBUG
struct CardIDLabel_Previews: PreviewProvider {
    static var previews: some View {
        Group {
            CardIDLabel(value: "34567453346")
                .environment(\.colorScheme, .light)

            CardIDLabel(value: "34567453346")
                .environment(\.colorScheme, .dark)
        }
        .previewLayout(
            .fixed(width: 300, height: 100)
        )
    }
}
#endif

```

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:03

Cards 〉 Cards 〉 CardIDLabel.swift 〉 No Selection

```swift
import SwiftUI

struct CardIDLabel: View {
    let value: String
    var body: some View {
        Text(self.value)
            .foregroundColor(.secondary)
    }
}

#if DEBUG
struct CardIDLabel_Previews: PreviewProvider {
    static var previews: some View {
        Group {
            CardIDLabel(value: "34567453346")
                .environment(\.colorScheme, .light)

            CardIDLabel(value: "34567453346")
                .environment(\.colorScheme, .dark)
        }
        .previewLayout(
            .fixed(width: 300, height: 100)
        )
    }
}
#endif
```

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:03

Cards 〉 Cards 〉 CardIDLabel.swift 〉 No Selection

```swift
import SwiftUI

struct CardIDLabel: View {
    let value: String
    var body: some View {
        Text(self.value)
            .foregroundColor(.secondary)
            .fontWeight(.bold)
    }
}

#if DEBUG
struct CardIDLabel_Previews: PreviewProvider {
    static var previews: some View {
        Group {
            CardIDLabel(value: "34567453346")
                .environment(\.colorScheme, .light)

            CardIDLabel(value: "34567453346")
                .environment(\.colorScheme, .dark)
        }
        .previewLayout(
            .fixed(width: 300, height: 100)
        )
    }
}
#endif
```

34567453346

Preview

34567453346

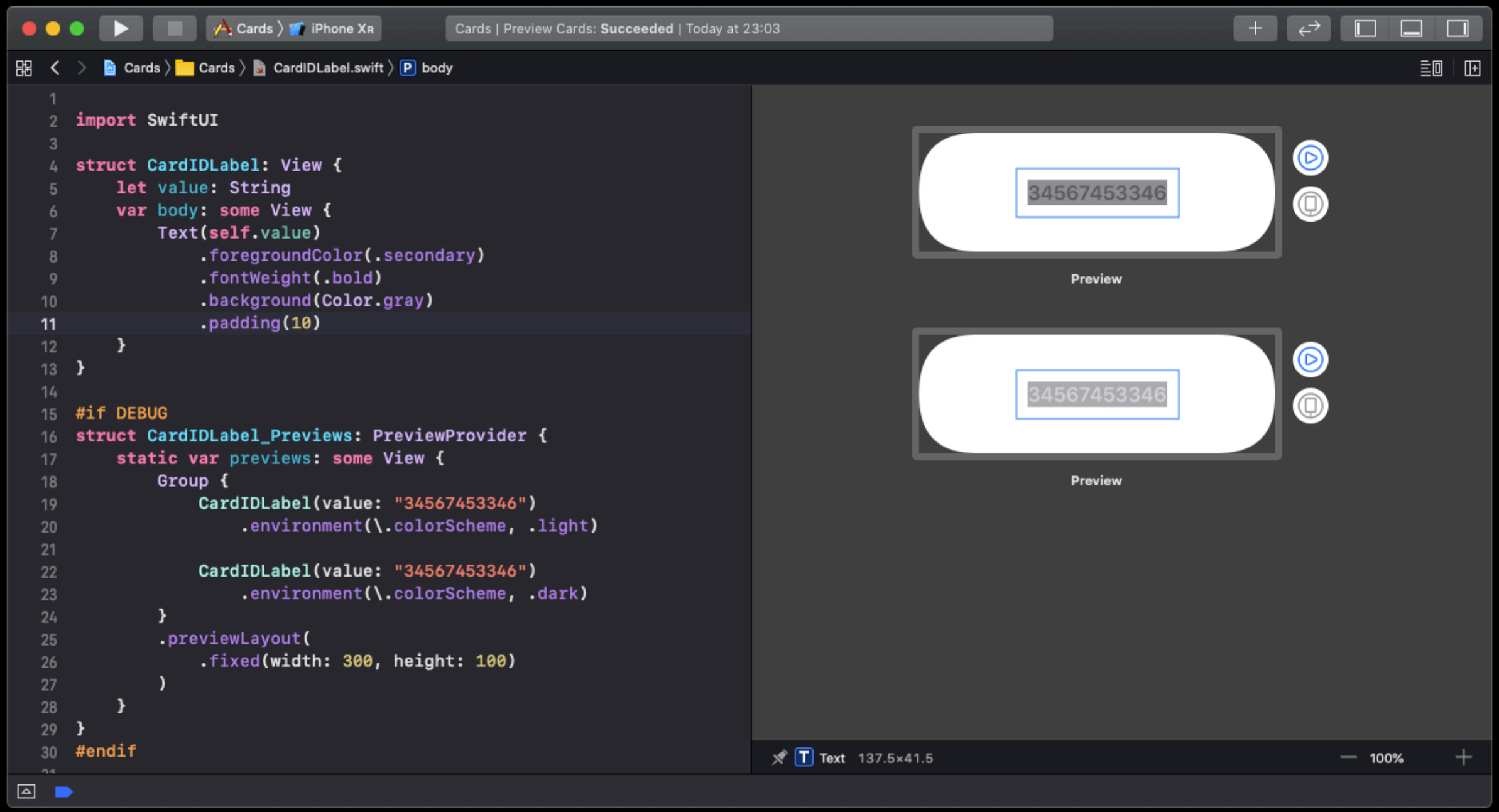
Preview

100%

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:03

Cards 〉 Cards 〉 CardIDLabel.swift 〉 No Selection

```swift

import SwiftUI

struct CardIDLabel: View {
    let value: String
    var body: some View {
        Text(self.value)
            .foregroundColor(.secondary)
            .fontWeight(.bold)
            .background(Color.gray)
    }
}

#if DEBUG
struct CardIDLabel_Previews: PreviewProvider {
    static var previews: some View {
        Group {
            CardIDLabel(value: "34567453346")
                .environment(\.colorScheme, .light)

            CardIDLabel(value: "34567453346")
                .environment(\.colorScheme, .dark)
        }
        .previewLayout(
            .fixed(width: 300, height: 100)
        )
    }
}
#endif

```


Cards 〉 iPhone XR
Cards | Preview Cards: Succeeded | Today at 23:03
Cards 〉 Cards 〉 CardIDLabel.swift 〉 body
import SwiftUI
struct CardIDLabel: View {
let value: String
var body: some View {
Text(self.value)
.foregroundColor(.secondary)
.fontWeight(.bold)
.background(Color.gray)
.padding(10)
}
}
#if DEBUG
struct CardIDLabel_Previews: PreviewProvider {
static var previews: some View {
Group {
CardIDLabel(value: "34567453346")
.environment(\.colorScheme, .light)
CardIDLabel(value: "34567453346")
.environment(\.colorScheme, .dark)
}
.previewLayout(
.fixed(width: 300, height: 100)
)
}
}
#endif
34567453346
Preview
34567453346
Preview
Text 137.5×41.5
100%

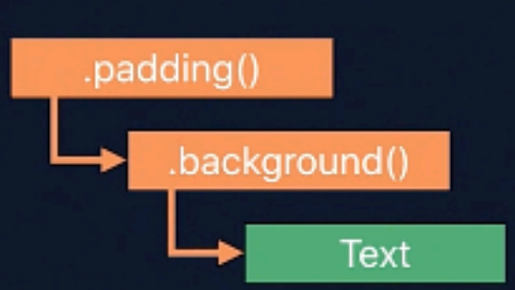
Modifiers

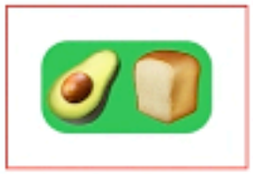
```
func foregroundColor(_ color: Color?) -> some View
```

```
func background<S>(_ content: StaticMember<S>) -> some View where S : Shape
    Style
```

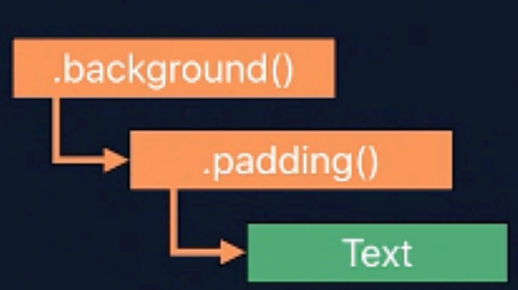
```
func padding(_ insets: EdgeInsets) -> some View
```

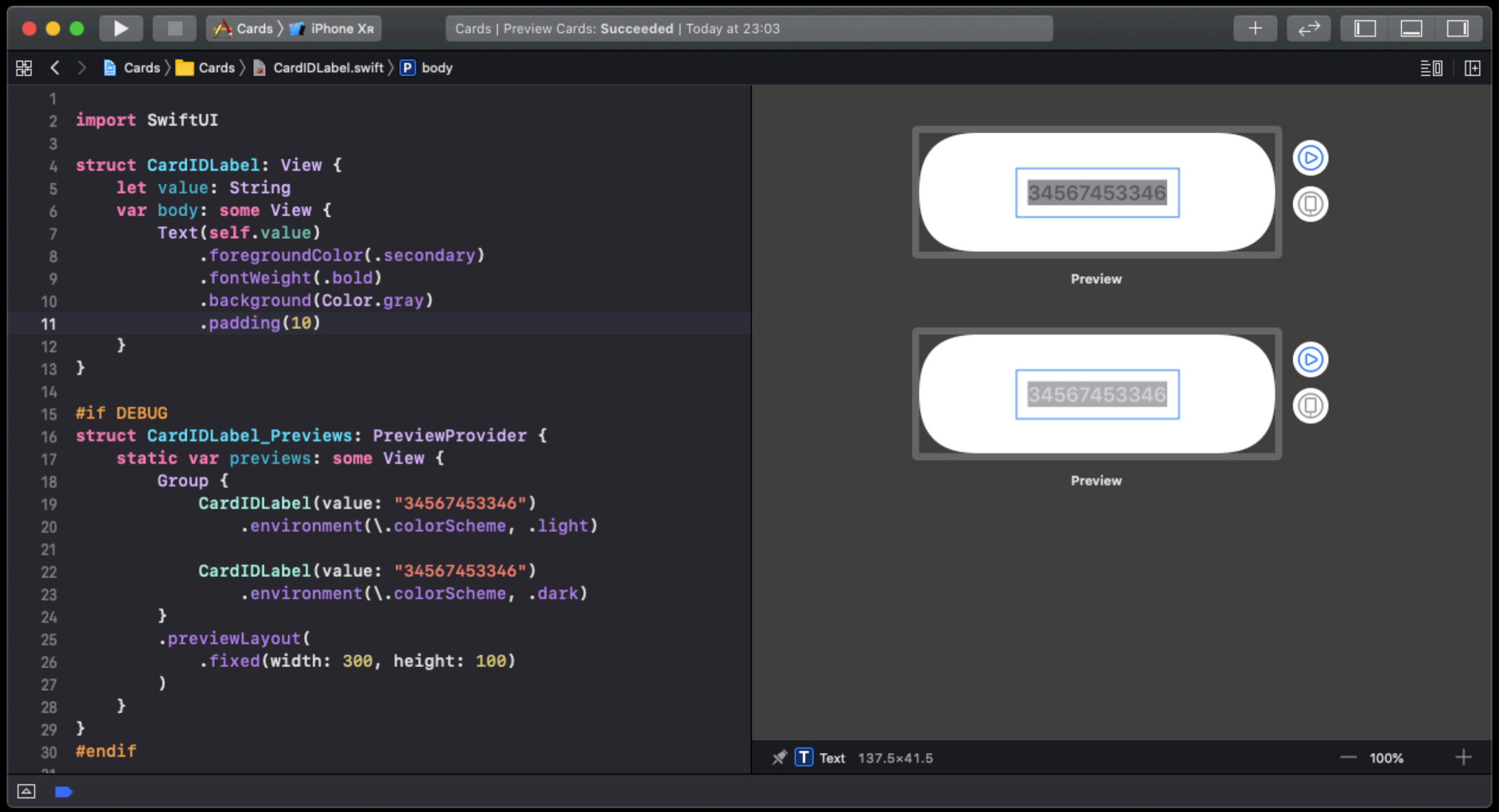
```swift
Text("🥑🍞")
    .background(Color.green, cornerRadius: 12)
    .padding(.all)
```

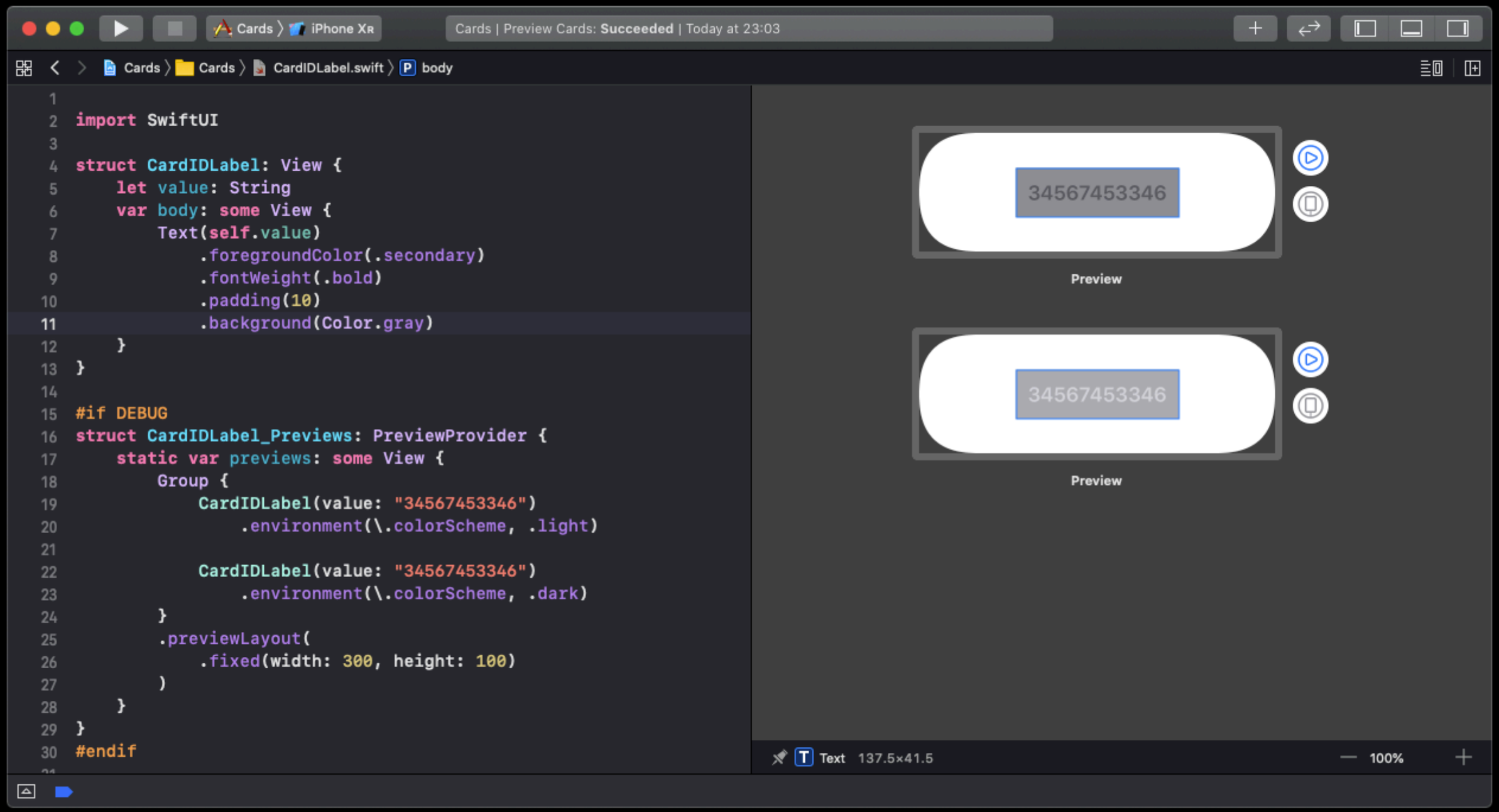




```swift
Text("🥑🍞")
    .padding(.all)
    .background(Color.green, cornerRadius: 12)
```




Cards 〉 iPhone XR
Cards | Preview Cards: Succeeded | Today at 23:03
Cards 〉 Cards 〉 CardIDLabel.swift 〉 body
import SwiftUI
struct CardIDLabel: View {
let value: String
var body: some View {
Text(self.value)
.foregroundColor(.secondary)
.fontWeight(.bold)
.background(Color.gray)
.padding(10)
}
}
#if DEBUG
struct CardIDLabel_Previews: PreviewProvider {
static var previews: some View {
Group {
CardIDLabel(value: "34567453346")
.environment(\.colorScheme, .light)
CardIDLabel(value: "34567453346")
.environment(\.colorScheme, .dark)
}
.previewLayout(
.fixed(width: 300, height: 100)
)
}
}
#endif
34567453346
Preview
34567453346
Preview
Text 137.5×41.5
100%


Cards 〉 iPhone XR
Cards | Preview Cards: Succeeded | Today at 23:03
Cards 〉 Cards 〉 CardIDLabel.swift 〉 body
import SwiftUI
struct CardIDLabel: View {
let value: String
var body: some View {
Text(self.value)
.foregroundColor(.secondary)
.fontWeight(.bold)
.padding(10)
.background(Color.gray)
}
}
#if DEBUG
struct CardIDLabel_Previews: PreviewProvider {
static var previews: some View {
Group {
CardIDLabel(value: "34567453346")
.environment(\.colorScheme, .light)
CardIDLabel(value: "34567453346")
.environment(\.colorScheme, .dark)
}
.previewLayout(
.fixed(width: 300, height: 100)
)
}
}
#endif
34567453346
Preview
34567453346
Preview
Text 137.5×41.5
100%

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:03

Cards 〉 Cards 〉 CardIDLabel.swift 〉 No Selection

```swift

import SwiftUI

struct CardIDLabel: View {
    let value: String
    var body: some View {
        Text(self.value)
            .foregroundColor(.secondary)
            .fontWeight(.bold)
            .padding(10)
            .background(Color.gray)
            .cornerRadius(20)
    }
}

#if DEBUG
struct CardIDLabel_Previews: PreviewProvider {
    static var previews: some View {
        Group {
            CardIDLabel(value: "34567453346")
                .environment(\.colorScheme, .light)

            CardIDLabel(value: "34567453346")
                .environment(\.colorScheme, .dark)
        }
        .previewLayout(
            .fixed(width: 300, height: 100)
        )
    }
}
#endif
```

34567453346

Preview

34567453346

Preview

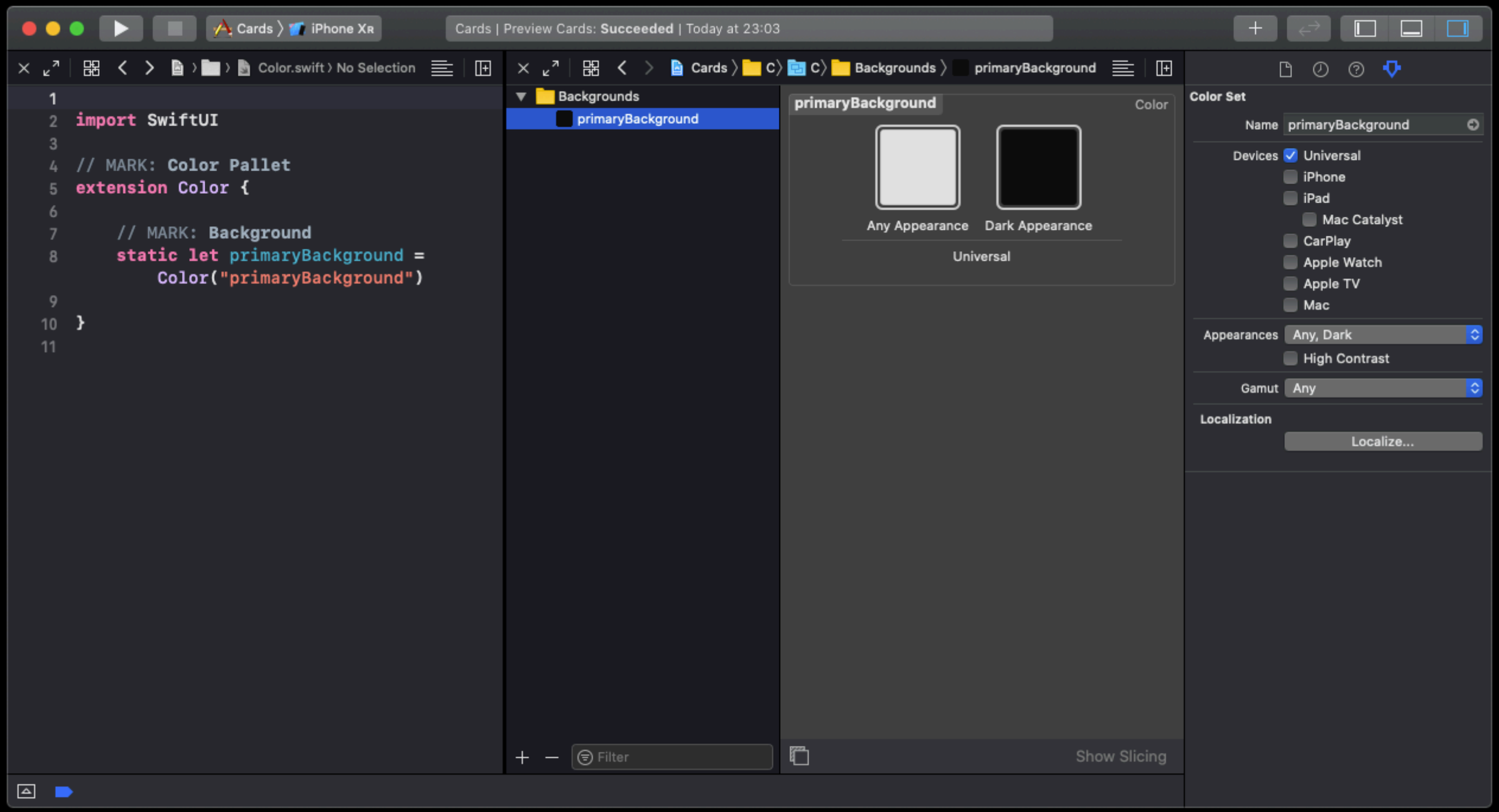
100%

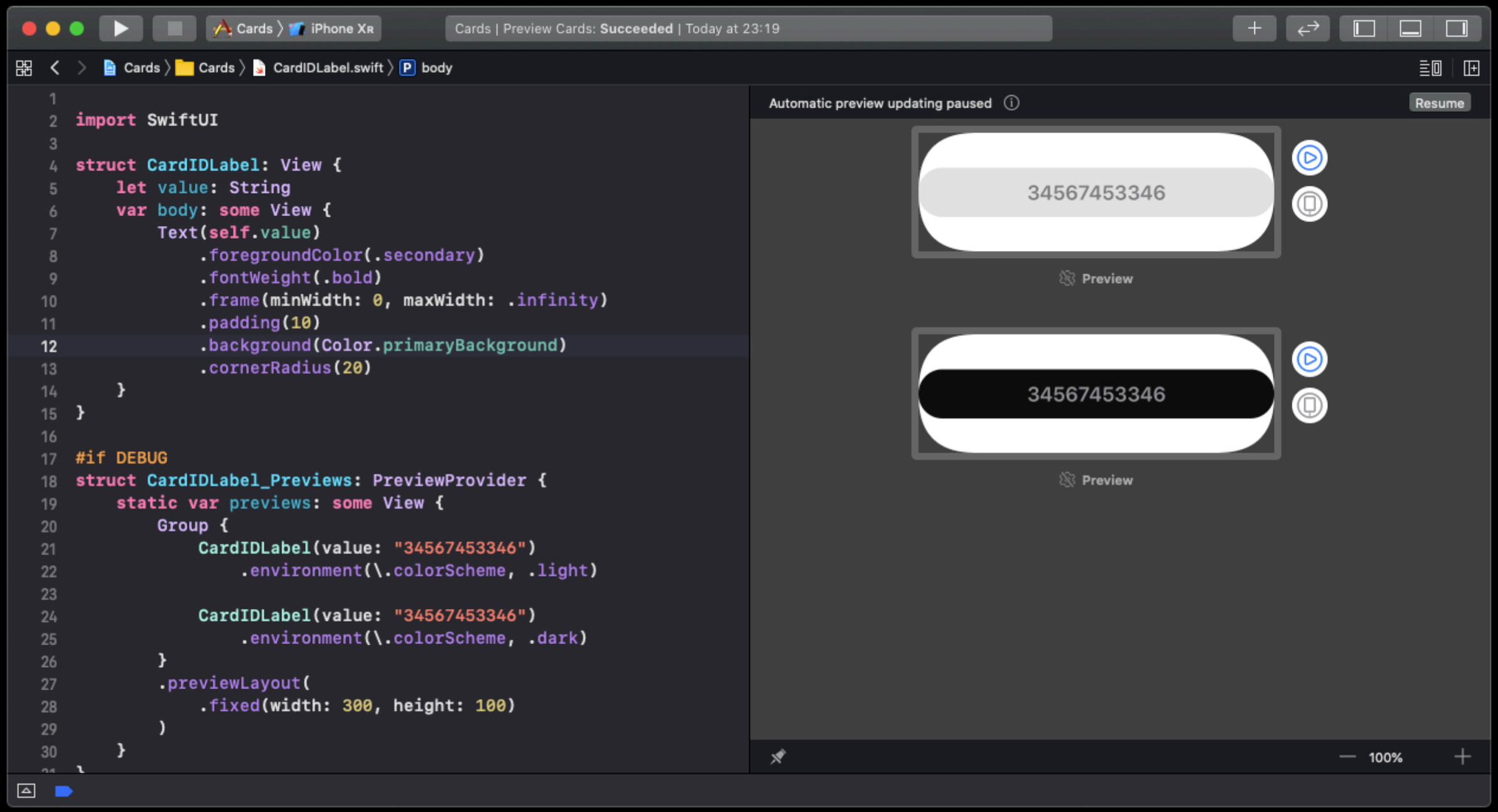
Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:03

Cards 〉 Cards 〉 CardIDLabel.swift 〉 No Selection

```swift

import SwiftUI

struct CardIDLabel: View {
    let value: String
    var body: some View {
        Text(self.value)
            .foregroundColor(.secondary)
            .fontWeight(.bold)
            .frame(minWidth: 0, maxWidth: .infinity)
            .padding(10)
            .background(Color.gray)
            .cornerRadius(20)
    }
}

#if DEBUG
struct CardIDLabel_Previews: PreviewProvider {
    static var previews: some View {
        Group {
            CardIDLabel(value: "34567453346")
                .environment(\.colorScheme, .light)

            CardIDLabel(value: "34567453346")
                .environment(\.colorScheme, .dark)
        }
        .previewLayout(
            .fixed(width: 300, height: 100)
        )
    }
}
```

Assets colors for dark mode



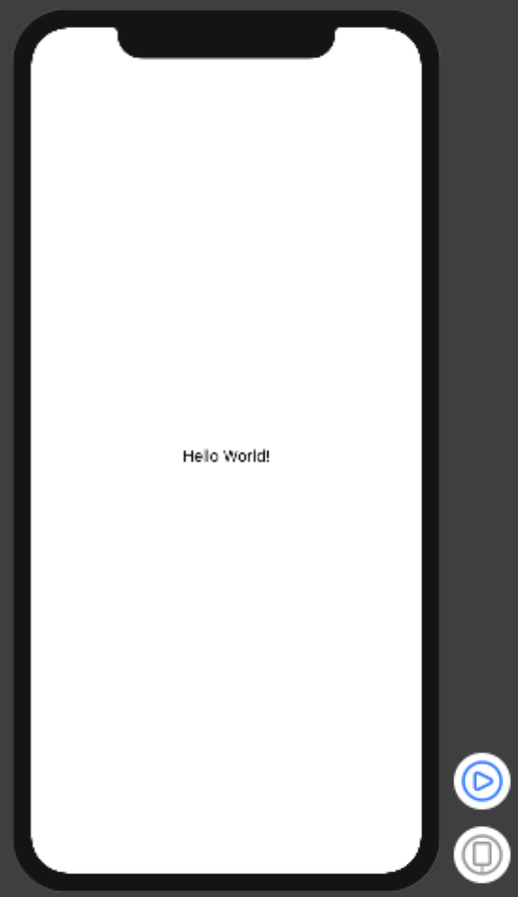

Cards 〉 iPhone XR
Cards | Preview Cards: Succeeded | Today at 23:19
Cards 〉 Cards 〉 CardIDLabel.swift 〉 body
1
2 import SwiftUI
3
4 struct CardIDLabel: View {
5 let value: String
6 var body: some View {
7 Text(self.value)
8 .foregroundColor(.secondary)
9 .fontWeight(.bold)
10 .frame(minWidth: 0, maxWidth: .infinity)
11 .padding(10)
12 .background(Color.primaryBackground)
13 .cornerRadius(20)
14 }
15 }
16
17 #if DEBUG
18 struct CardIDLabel_Previews: PreviewProvider {
19 static var previews: some View {
20 Group {
21 CardIDLabel(value: "34567453346")
22 .environment(\.colorScheme, .light)
23
24 CardIDLabel(value: "34567453346")
25 .environment(\.colorScheme, .dark)
26 }
27 .previewLayout(
28 .fixed(width: 300, height: 100)
29)
30 }
Automatic preview updating paused
Resume
34567453346
Preview
34567453346
Preview
100%

Cards 〉 iPhone XR

Cards | Preview Cards: **Succeeded** | Today at 23:25

Cards 〉 Cards 〉 Card.swift 〉 No Selection

```swift

import SwiftUI

struct CardView: View {
    var body: some View {
       Text("Hello World!")
    }
}

#if DEBUG
struct CardView_Previews : PreviewProvider {
    static var previews: some View {
        CardView()
    }
}
#endif

```



Preview

50%

Cards 〉 iPhone XR

Cards | Preview Cards: **Succeeded** | Today at 23:25

Cards 〉 Cards 〉 Card.swift 〉 No Selection

```swift
import SwiftUI

struct CardView: View {
    var body: some View {
        Text("Hello World!")
    }
}

#if DEBUG
struct CardView_Previews : PreviewProvider {
    static var previews: some View {
        Group {
            CardView()
        }
        .previewLayout(.fixed(width: 400, height: 400))
    }
}
#endif
```

Hello World!

Preview

75%

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:27

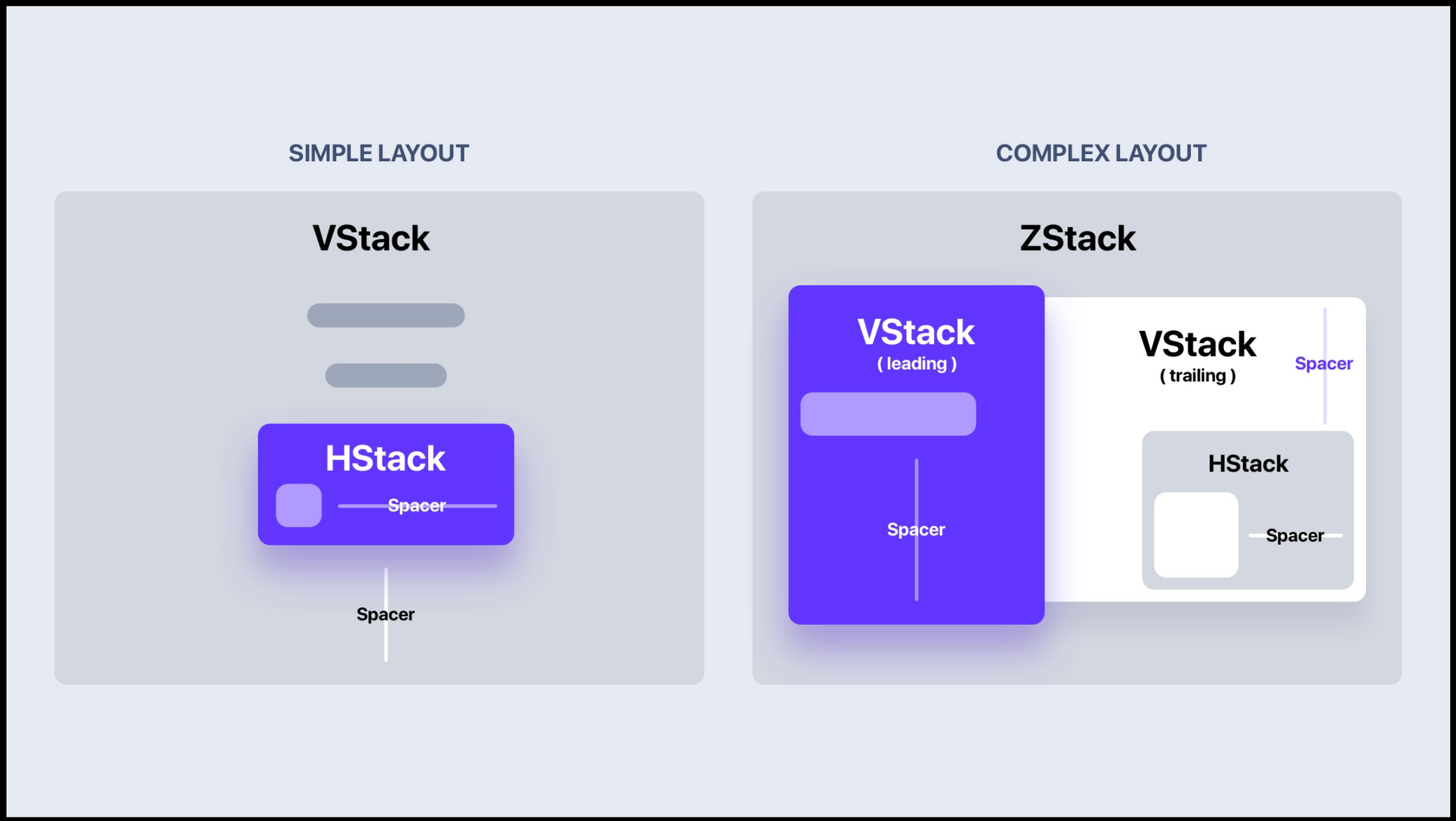
Cards 〉 Cards 〉 Card.swift 〉 No Selection

```swift
import SwiftUI

struct CardView: View {

    var body: some View {
        Image("bip_logo")
    }
}

#if DEBUG
struct CardView_Previews : PreviewProvider {
    static var previews: some View {
        Group {
            CardView()
        }
        .previewLayout(.fixed(width: 400, height: 400))
    }
}
#endif
```

Preview

75%

Stacks and Spacer

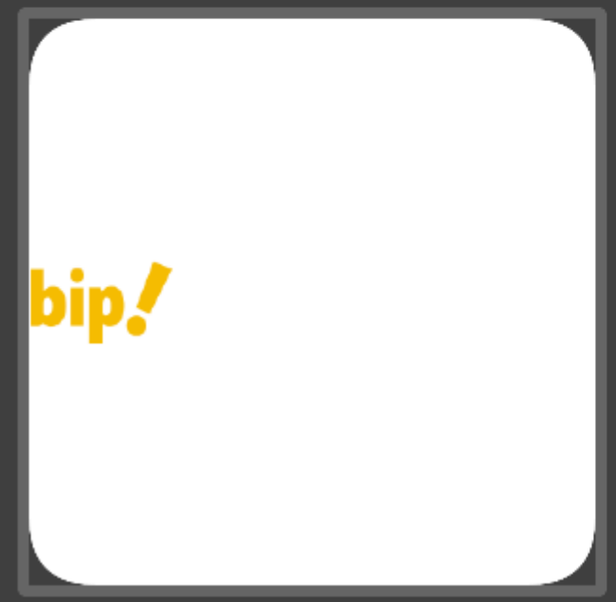


*Por defecto todos los views intentaran ocupar el menor espacio

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:27

Cards 〉 Cards 〉 Card.swift 〉 No Selection

```swift
import SwiftUI

struct CardView: View {

    var body: some View {
        HStack(alignment: .top) {
            Image("bip_logo")
            Spacer()
        }
    }
}

#if DEBUG
struct CardView_Previews : PreviewProvider {
    static var previews: some View {
        Group {
            CardView()
        }
        .previewLayout(.fixed(width: 400, height: 400))
    }
}
#endif

```



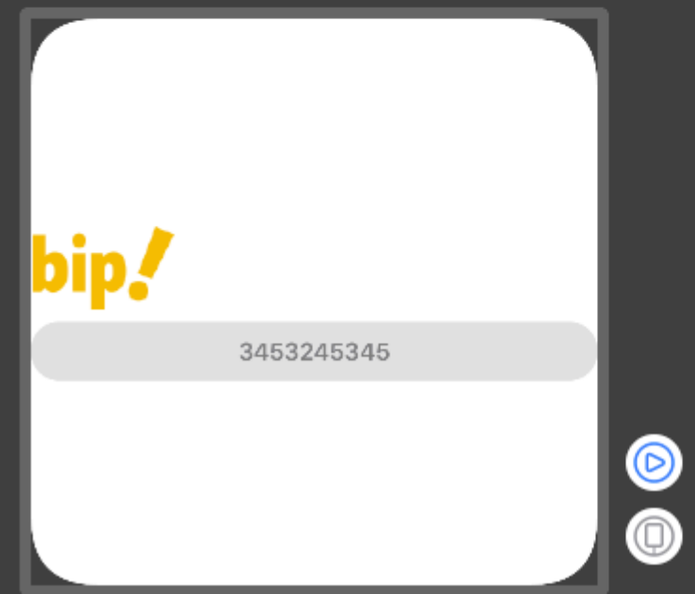
Preview

75%

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:27

Cards 〉 Cards 〉 Card.swift 〉 No Selection

```swift

import SwiftUI

struct CardView: View {

    var body: some View {
        VStack {
            HStack(alignment: .top) {
                Image("bip_logo")
                Spacer()
            }

            CardIDLabel(value: "3453245345")
        }
    }
}

#if DEBUG
struct CardView_Previews : PreviewProvider {
    static var previews: some View {
        Group {
            CardView()
        }
        .previewLayout(.fixed(width: 400, height: 400))
    }
}
#endif

```



Preview

75%

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:30

Cards 〉 Cards 〉 Card.swift 〉 No Selection

```swift

import SwiftUI

struct CardView: View {

    var body: some View {
            VStack {
                HStack(alignment: .top) {
                    Image("bip_logo")
                    Spacer()
                }

                Spacer()

                CardIDLabel(value: "3453245345")
            }
    }
}

#if DEBUG
struct CardView_Previews : PreviewProvider {
    static var previews: some View {
        Group {
            CardView()
        }
        .previewLayout(.fixed(width: 400, height: 400))
    }
}
#endif

```

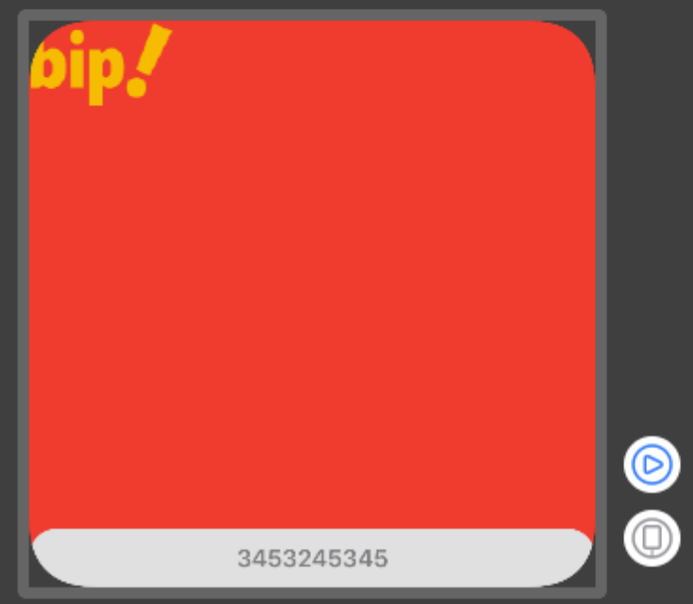
Preview

75%

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:30

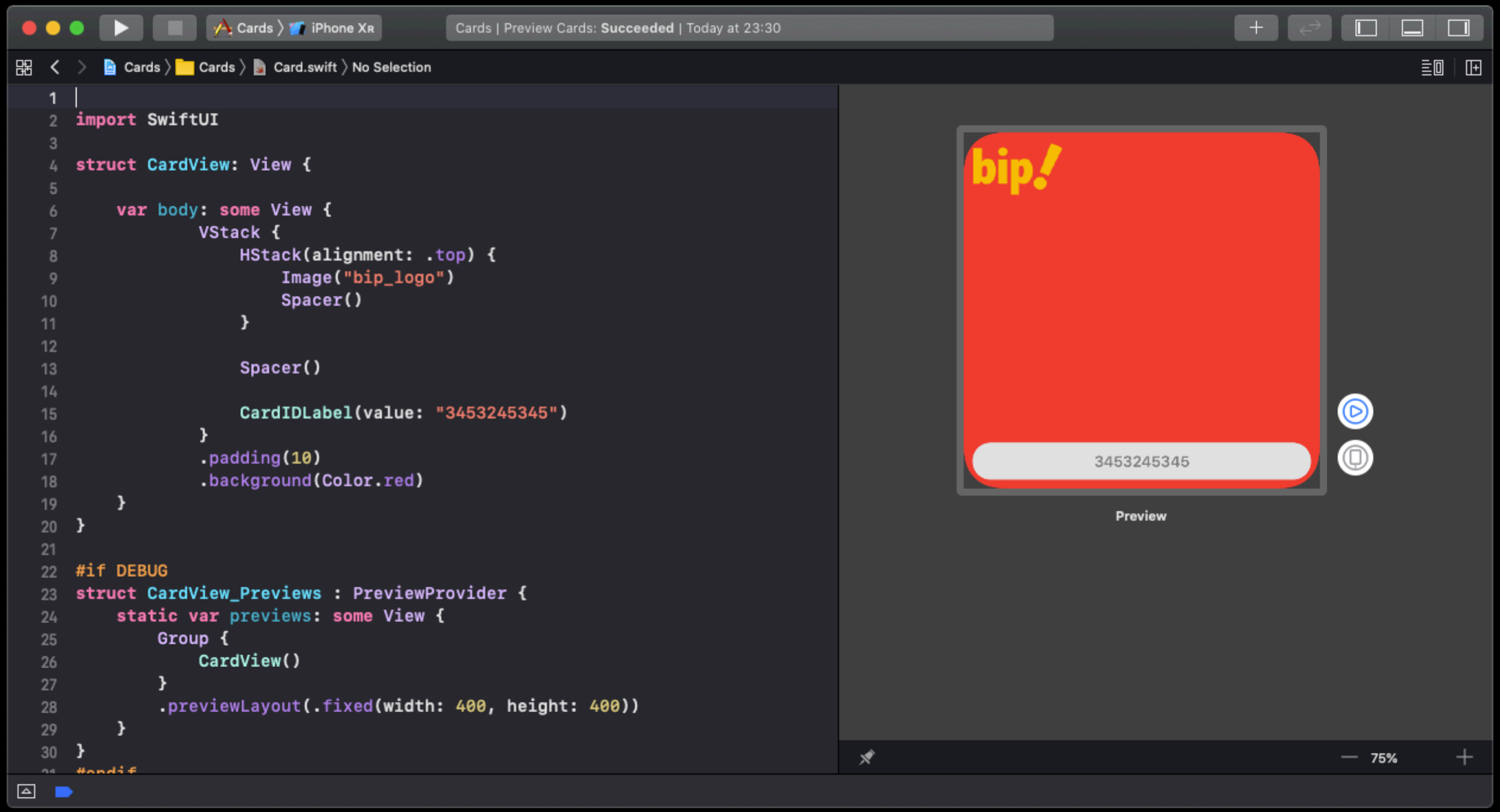
Cards 〉 Cards 〉 Card.swift 〉 No Selection

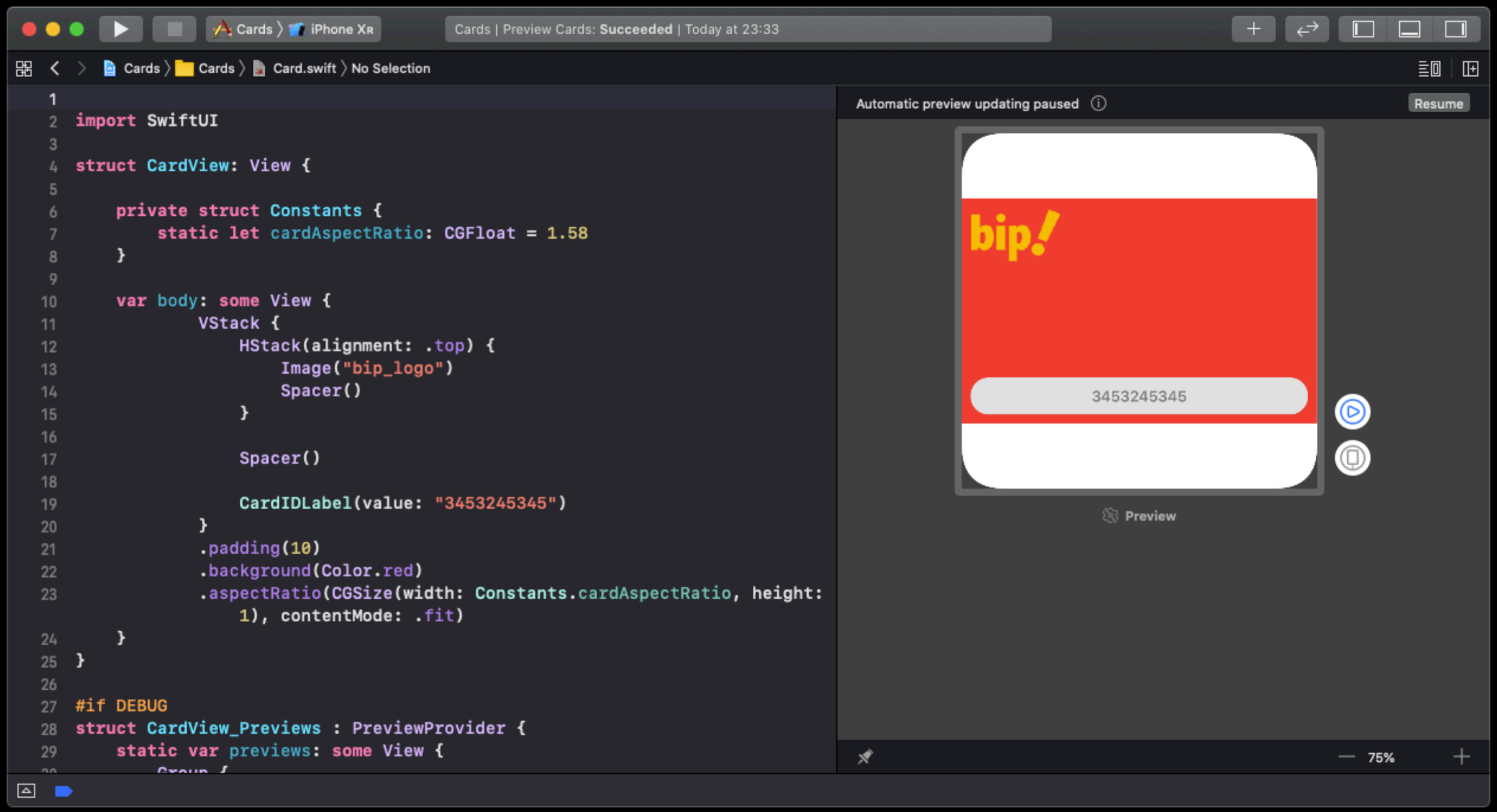
```swift
import SwiftUI

struct CardView: View {

    var body: some View {
            VStack {
                HStack(alignment: .top) {
                    Image("bip_logo")
                    Spacer()
                }

                Spacer()

                CardIDLabel(value: "3453245345")
            }
            .background(Color.red)
    }
}

#if DEBUG
struct CardView_Previews : PreviewProvider {
    static var previews: some View {
        Group {
            CardView()
        }
        .previewLayout(.fixed(width: 400, height: 400))
    }
}
#endif
```



Preview

75%


Cards 〉 iPhone XR
Cards | Preview Cards: Succeeded | Today at 23:30
Cards 〉 Cards 〉 Card.swift 〉 No Selection
1
2 import SwiftUI
3
4 struct CardView: View {
5
6 var body: some View {
7 VStack {
8 HStack(alignment: .top) {
9 Image("bip_logo")
10 Spacer()
11 }
12
13 Spacer()
14
15 CardIDLabel(value: "3453245345")
16 }
17 .padding(10)
18 .background(Color.red)
19 }
20 }
21
22 #if DEBUG
23 struct CardView_Previews : PreviewProvider {
24 static var previews: some View {
25 Group {
26 CardView()
27 }
28 .previewLayout(.fixed(width: 400, height: 400))
29 }
30 }
bip!
3453245345
Preview
75%

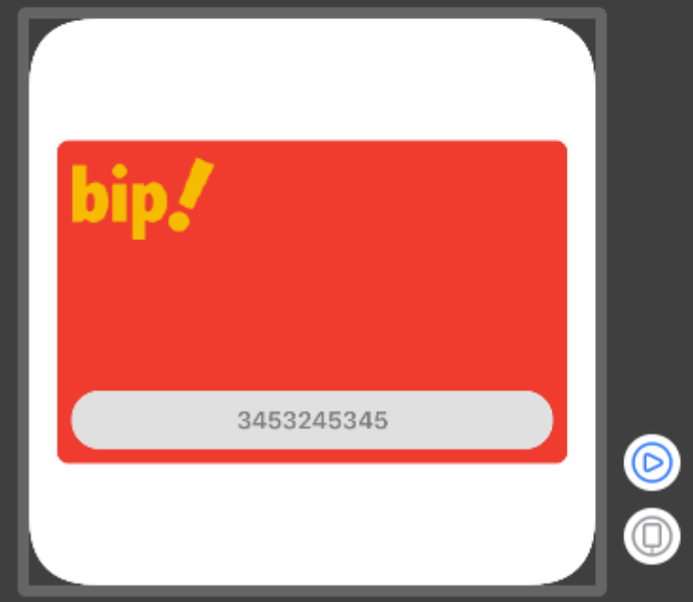

Cards 〉 iPhone XR
Cards | Preview Cards: Succeeded | Today at 23:33
Cards 〉 Cards 〉 Card.swift 〉 No Selection
Automatic preview updating paused
Resume
import SwiftUI
struct CardView: View {
private struct Constants {
static let cardAspectRatio: CGFloat = 1.58
}
var body: some View {
VStack {
HStack(alignment: .top) {
Image("bip_logo")
Spacer()
}
Spacer()
CardIDLabel(value: "3453245345")
}
.padding(10)
.background(Color.red)
.aspectRatio(CGSize(width: Constants.cardAspectRatio, height:
1), contentMode: .fit)
}
}
#if DEBUG
struct CardView_Previews : PreviewProvider {
static var previews: some View {
bip!
3453245345
Preview
75%

Cards 〉 iPhone XR

Cards | Preview Cards: **Succeeded** | Today at 23:35

Cards 〉 Cards 〉 Card.swift 〉 No Selection

```swift
import SwiftUI

struct CardView: View {

    private struct Constants {
        static let cardAspectRatio: CGFloat = 1.58
        static let cornerRadius: CGFloat = 8
    }

    var body: some View {
            VStack {
                HStack(alignment: .top) {
                    Image("bip_logo")
                    Spacer()
                }

                Spacer()

                CardIDLabel(value: "3453245345")
            }
            .padding(10)
            .background(Color.red)
            .aspectRatio(CGSize(width: Constants.cardAspectRatio, height:
                1), contentMode: .fit)
            .cornerRadius(Constants.cornerRadius)
    }
}

#if DEBUG
```



Preview

75%

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:35

Cards 〉 Cards 〉 Card.swift 〉 No Selection

```swift
import SwiftUI

struct CardView: View {

    private struct Constants {
        static let cardAspectRatio: CGFloat = 1.58
        static let cornerRadius: CGFloat = 8
    }

    var body: some View {
            VStack {
                HStack(alignment: .top) {
                    Image("bip_logo")
                    Spacer()
                }

                Spacer()

                CardIDLabel(value: "3453245345")
            }
            .padding(10)
            .background(
                Rectangle()
                .fill(
                    LinearGradient(
                        gradient: Gradient(colors: [Color.red, Color.blue]),
                        startPoint: .top,
                        endPoint: .bottom
                    )
                )
                .cornerRadius(Constants.cornerRadius)
            )
            .aspectRatio(CGSize(width: Constants.cardAspectRatio, height: 1), contentMode: .fit)
            .cornerRadius(Constants.cornerRadius)
    }
}

#if DEBUG
struct CardView_Previews : PreviewProvider {
    static var previews: some View {
```

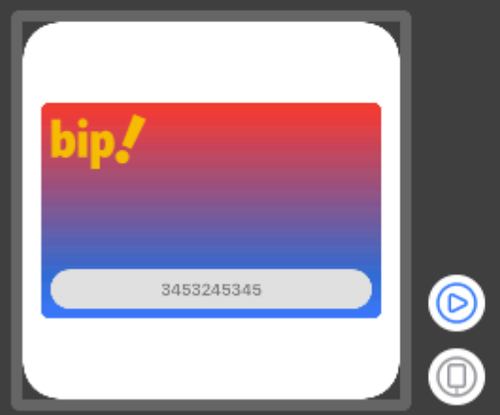
Preview

75%

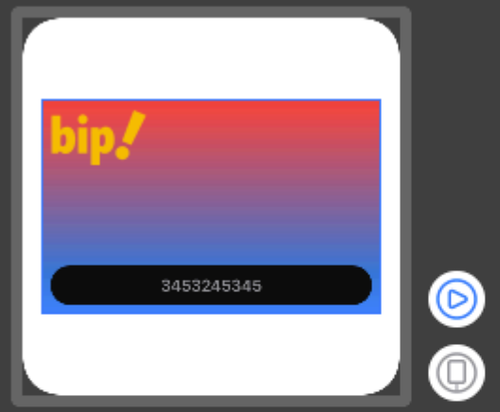
Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:35

Cards 〉 Cards 〉 Card.swift 〉 previews

```
            .padding(10)
            .background(
                Rectangle()
                .fill(
                    LinearGradient(
                        gradient: Gradient(colors: [Color.red, Color.blue]),
                        startPoint: .top,
                        endPoint: .bottom
                    )
                )
                .cornerRadius(Constants.cornerRadius)
            )
            .aspectRatio(CGSize(width: Constants.cardAspectRatio, height: 1), contentMode: .fit)
            .cornerRadius(Constants.cornerRadius)
    }
}

#if DEBUG
struct CardView_Previews : PreviewProvider {
    static var previews: some View {
        Group {
            CardView()
            .environment(\.colorScheme, .light)

            CardView()
            .environment(\.colorScheme, .dark)
        }
        .padding(20)
        .previewLayout(.fixed(width: 400, height: 400))
    }
}
#endif
```



Preview



Preview

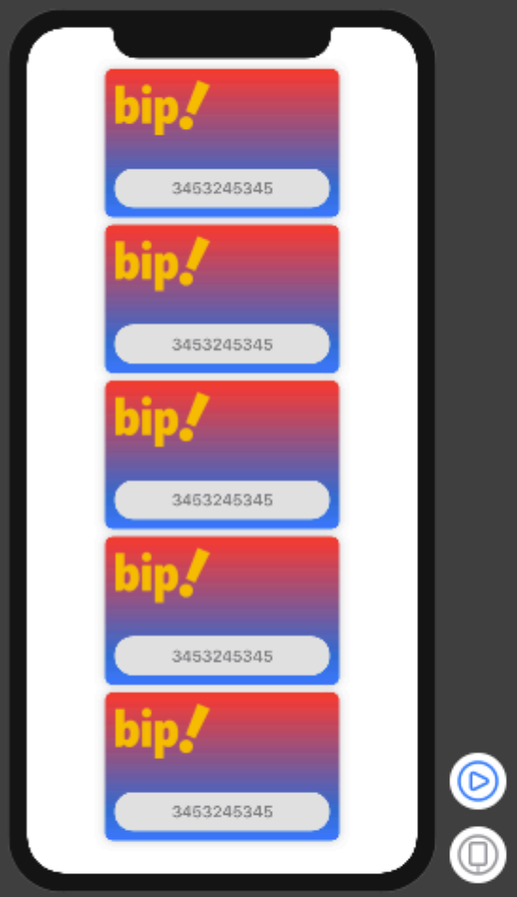
CardView 360×228 — 50% +

CardsStack

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:43

Cards 〉 Cards 〉 CardsStackView.swift 〉 No Selection

```swift

import SwiftUI

struct CardsStackView: View {
    var body: some View {
        VStack {
            CardView()
            CardView()
            CardView()
            CardView()
            CardView()
        }
    }
}

#if DEBUG
struct CardsStackView_Previews: PreviewProvider {
    static var previews: some View {
        CardsStackView()
    }
}
#endif

```



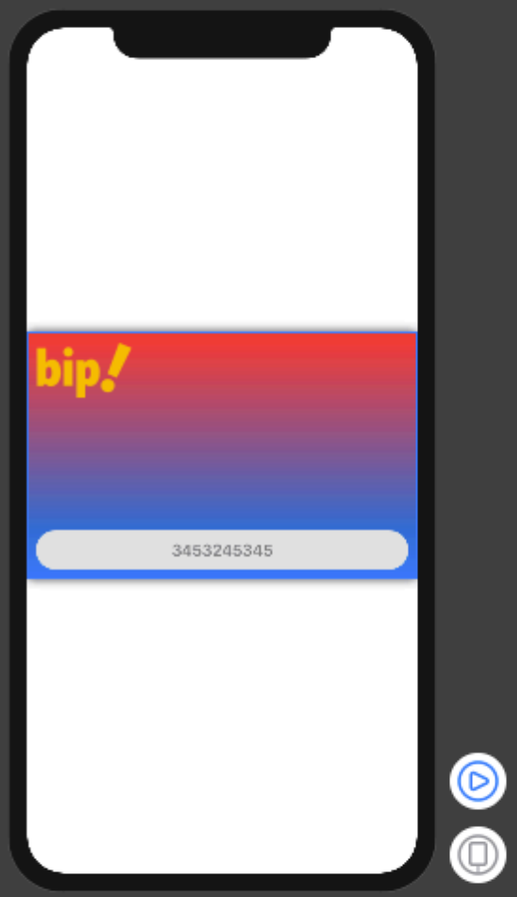
Preview

50%

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:43

Cards 〉 Cards 〉 CardsStackView.swift 〉 P body

```swift

import SwiftUI

struct CardsStackView: View {
    var body: some View {
        ZStack {
            CardView()
            CardView()
            CardView()
            CardView()
            CardView()
        }
    }
}

#if DEBUG
struct CardsStackView_Previews: PreviewProvider {
    static var previews: some View {
        CardsStackView()
    }
}
#endif

```



Preview

ZStack 414×262 — 50% +

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:45

Cards 〉 Cards 〉 CardsStackView.swift 〉 No Selection

```swift

import SwiftUI

struct CardsStackView: View {
    var body: some View {
        ZStack {
            CardView()
            CardView()
            CardView()
            CardView()
            CardView()
        }
    }
}

#if DEBUG
struct CardsStackView_Previews : PreviewProvider {
    static var previews: some View {
        Group {
            CardsStackView()
        }
        .padding(20)
        .previewLayout(.fixed(width: 400, height: 400))
    }
}
#endif


```

Preview

75%

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:45

Cards 〉 Cards 〉 CardsStackView.swift 〉 No Selection

```swift

import SwiftUI

struct CardsStackView: View {
    var body: some View {
        ZStack {
            CardView()
                .opacity(0.5)
            CardView()
                .opacity(0.5)
            CardView()
                .opacity(0.5)
            CardView()
                .opacity(0.5)
        }
    }
}

#if DEBUG
struct CardsStackView_Previews : PreviewProvider {
    static var previews: some View {
        Group {
            CardsStackView()
        }
        .padding(20)
        .previewLayout(.fixed(width: 400, height: 400))
    }
}
#endif

```

Preview

75%

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 23:45

Cards 〉 Cards 〉 CardsStackView.swift 〉 No Selection

```swift
import SwiftUI

struct CardsStackView: View {
    var body: some View {
        ZStack {
            CardView()
                .scaleEffect(0.7)
                .opacity(0.5)
            CardView()
                .scaleEffect(0.8)
                .opacity(0.5)
            CardView()
                .scaleEffect(0.9)
                .opacity(0.5)
            CardView()
                .scaleEffect(1)
                .opacity(0.5)
        }
    }
}

#if DEBUG
struct CardsStackView_Previews : PreviewProvider {
    static var previews: some View {
        Group {
            CardsStackView()
        }
        .padding(20)
        .previewLayout(.fixed(width: 400, height: 400))
    }
```

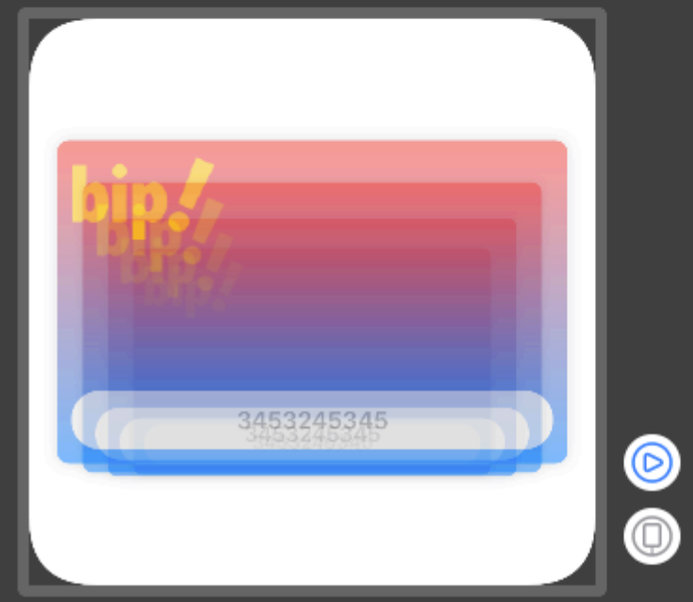
Preview

75%

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Yesterday at 23:45

Cards 〉 Cards 〉 CardsStackView.swift 〉 No Selection

```swift

import SwiftUI

struct CardsStackView: View {
    var body: some View {
        ZStack {
            CardView()
                .offset(y: 60)
                .scaleEffect(0.7)
                .opacity(0.5)
            CardView()
                .offset(y: 40)
                .scaleEffect(0.8)
                .opacity(0.5)
            CardView()
                .offset(y: 20)
                .scaleEffect(0.9)
                .opacity(0.5)
            CardView()
                .offset(y: 0)
                .scaleEffect(1)
                .opacity(0.5)
        }
    }
}

#if DEBUG
struct CardsStackView_Previews : PreviewProvider {
    static var previews: some View {
        Group {
```



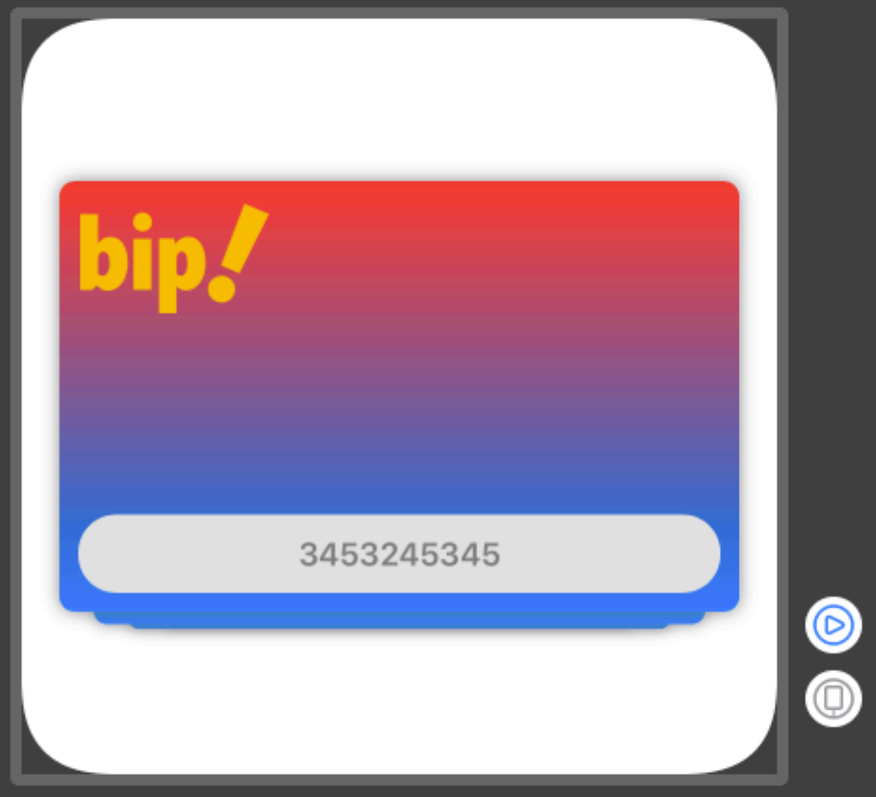
Preview

75%

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 0:03

Cards 〉 Cards 〉 CardsStackView.swift 〉 No Selection

```swift

import SwiftUI

struct CardsStackView: View {
    var body: some View {
        ZStack {
            CardView()
                .offset(y: 60)
                .scaleEffect(0.7)
            CardView()
                .offset(y: 40)
                .scaleEffect(0.8)
            CardView()
                .offset(y: 20)
                .scaleEffect(0.9)
            CardView()
                .offset(y: 0)
                .scaleEffect(1)
        }
    }
}

#if DEBUG
struct CardsStackView_Previews : PreviewProvider {
    static var previews: some View {
        Group {
            CardsStackView()
        }
        .padding(20)
        .previewLayout(.fixed(width: 400, height: 400))
    }
```



Preview

100%

Identifiable

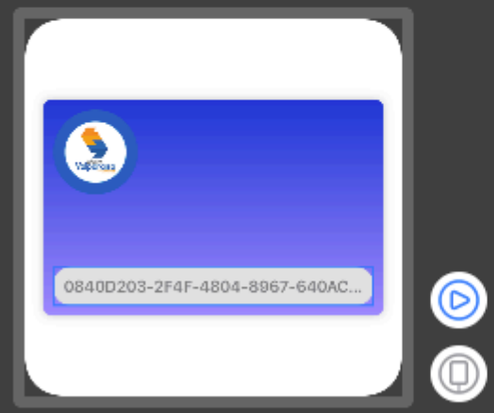
```swift
struct Card: Identifiable {
    let id: String
    let balance: String
    let vendor: CardVendor
}

enum CardVendor: CaseIterable {
    case bip
    case valpo

    var description: String { ••• }
    var image: String { ••• }
    var colorTop: Color { ••• }
    var colorBottom: Color { ••• }
}
```

```swift
#if DEBUG
extension Card {
    static let previewContent: [Card] = [
        Card(
            id: UUID().uuidString,
            balance: "1000",
            vendor: .bip
        ),
        Card(
            id: UUID().uuidString,
            balance: "1500",
            vendor: .valpo
        ),
        Card(
            id: UUID().uuidString,
            balance: "2000",
            vendor: .bip
        ),
        Card(
            id: UUID().uuidString,
            balance: "500",
            vendor: .valpo
        )
    ]
}
#endif
```

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 21:36

Cards 〉 Cards 〉 Card.swift 〉 P body

```
40
41  #if DEBUG
42  struct CardView_Previews : PreviewProvider {
43      static var previews: some View {
44          Group {
45              CardView(card: Card.previewContent[1])
46              .scaledToFit()
47              .environment(\.colorScheme, .light)
48
49              CardView(card: Card.previewContent[0])
50                  .scaledToFit()
51                  .environment(\.colorScheme, .dark)
52          }
53          .padding(20)
54          .previewLayout(.fixed(width: 400, height: 400))
55      }
56  }
57  #endif
58
```

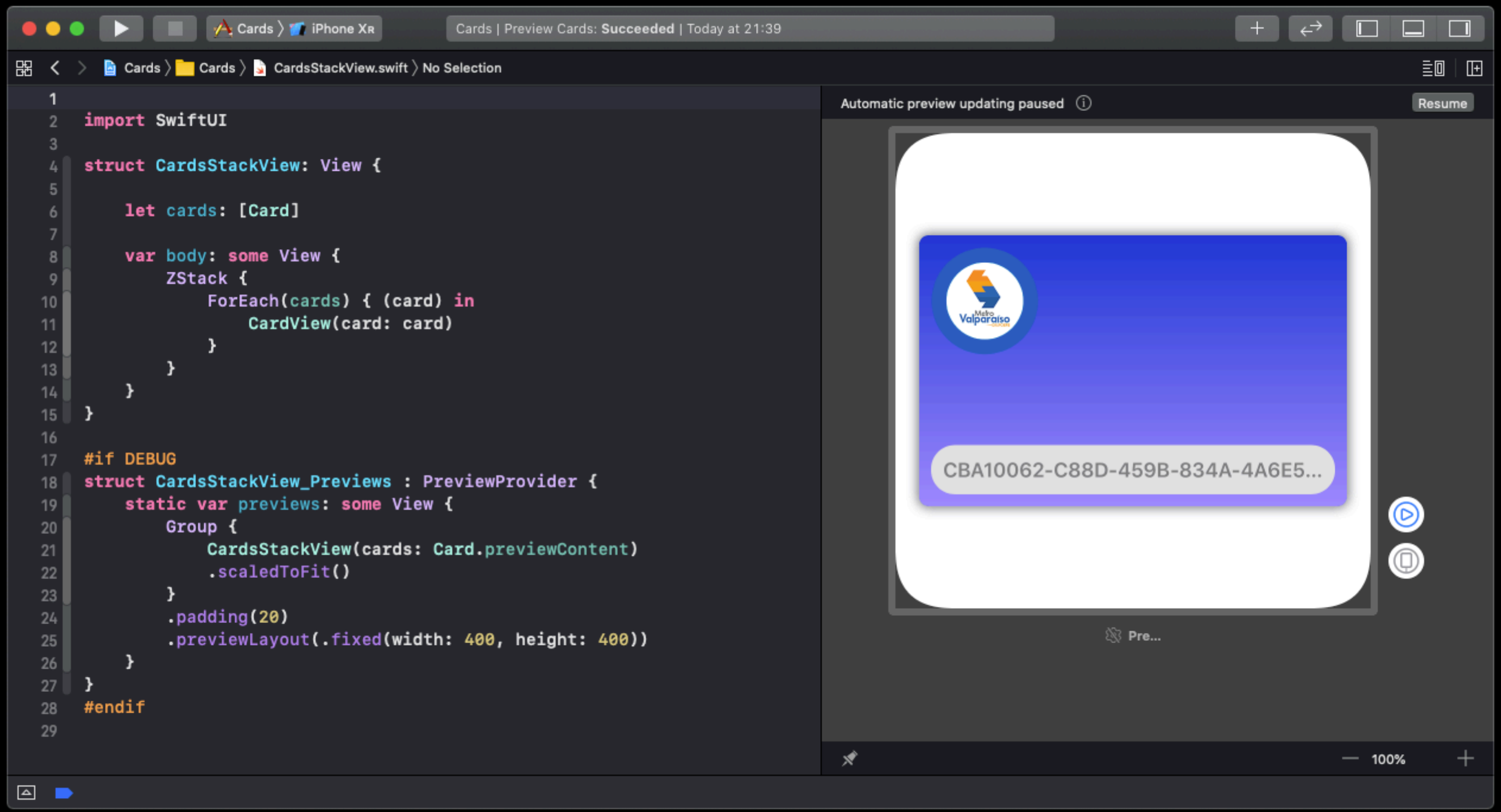


Preview

Preview

C CardIDLabel 340×41.5 — 50% +

ForEach




```swift

import SwiftUI

struct CardsStackView: View {

    let cards: [Card]

    var body: some View {
        ZStack {
            ForEach(cards) { (card) in
                CardView(card: card)
            }
        }
    }
}

#if DEBUG
struct CardsStackView_Previews : PreviewProvider {
    static var previews: some View {
        Group {
            CardsStackView(cards: Card.previewContent)
            .scaledToFit()
        }
        .padding(20)
        .previewLayout(.fixed(width: 400, height: 400))
    }
}
#endif

```

```swift
private func getIndex(for card: Card) -> Int {
    return cards.firstIndex(where: { $0.id == card.id })!
}

private func cardScale(forIndex index: Int, totalCards: Int) -> CGFloat {
    guard totalCards > 1 else { return 1 }

    // Line Ecuation: x = (-y + a) / a
    let scaleFactor: Float = Float(-index + totalCards - 1) / Float(totalCards - 1)
    let scaleOffset = Constants.cardsScaleOffset * scaleFactor

    return CGFloat(1 - scaleOffset)
}

private func yPossition(forIndex index: Int, totalCards: Int) -> CGFloat {
    guard totalCards > 1 else { return 0 }

    // Line Ecuation: y = (x - a) * b
    let yPos: Float = Float(index - (totalCards - 1)) * -Constants.cardsVerticalOffset
    return CGFloat(yPos)
}
```

Cards 〉 iPhone XR

Cards | Preview Cards: **Succeeded** | Today at 22:08

Cards 〉 Cards 〉 CardsStackView.swift 〉 CardsStackView

```swift

    private struct Constants {
        static let cardsVerticalOffset: Float = 15
        static let cardsScaleOffset: Float = 0.2
    }

    let cards: [Card]

    var body: some View {
        ZStack {
            ForEach(cards) { (card) in
                CardView(card: card)
                    .scaleEffect(self.cardScale(
                        forIndex: self.getIndex(for: card),
                        totalCards: self.cards.count)
                    )
                    .offset(y: self.yPossition(
                        forIndex: self.getIndex(for: card),
                        totalCards: self.cards.count)
                    )
            }
        }
    }

    private func getIndex(for card: Card) -> Int {
        return cards.firstIndex(where: { $0.id == card.id })!
    }

    private func cardScale(forIndex index: Int, totalCards: Int) -> CGFloat {
        guard totalCards > 1 else { return 1 }

        // Line Ecuation: x = (-y + a) / a
        let scaleFactor: Float = Float(-index + totalCards - 1) / Float(totalCards - 1)
        let scaleOffset = Constants.cardsScaleOffset * scaleFactor
```

Preview

100%

Custom Modifiers

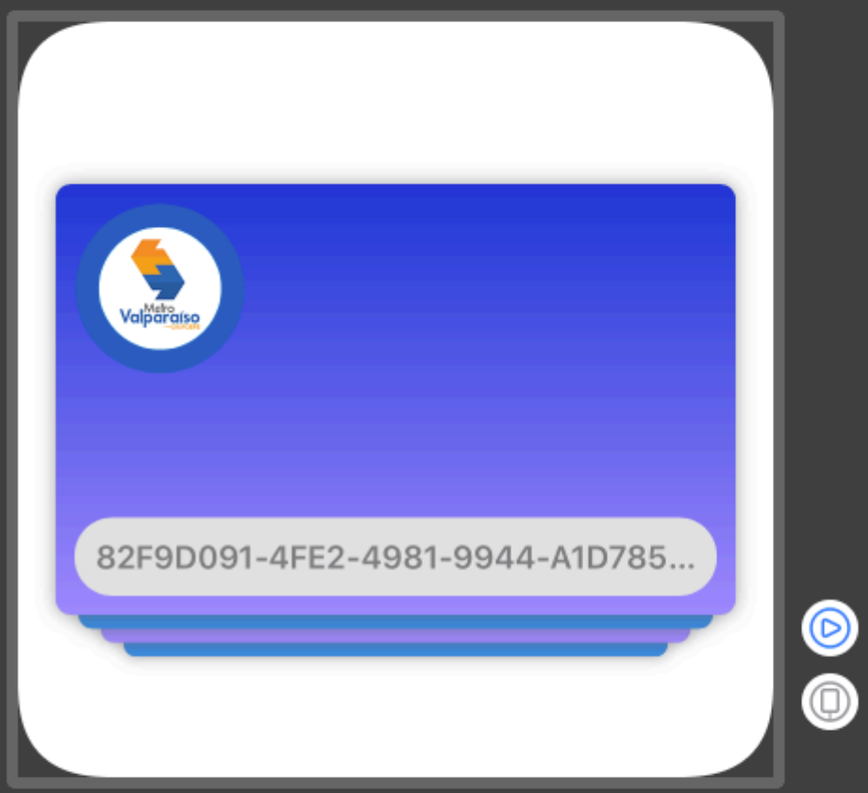
```swift

import SwiftUI

struct CardStyle {
    let scale: CGFloat
    let yPos: CGFloat
}

struct CardStyleModifier: ViewModifier {
    let style: CardStyle

    func body(content: Content) -> some View {
        content
            .scaleEffect(style.scale)
            .offset(y: style.yPos)
    }
}

```

```
private func style(for card: Card) -> CardStyle {
    let totalCards = cards.count
    let index = getIndex(for: card)

    let scale = cardScale(
        forIndex: index,
        totalCards: totalCards
    )

    let yPos = yPossition(
        forIndex: index,
        totalCards: totalCards
    )

    return CardStyle(scale: scale,
                     yPos: yPos)
}
```

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 22:08

Cards 〉 Cards 〉 CardsStackView.swift 〉 M getIndex(for:)

```swift

import SwiftUI

struct CardsStackView: View {

    private struct Constants {
        static let cardsVerticalOffset: Float = 15
        static let cardsScaleOffset: Float = 0.2
    }

    let cards: [Card]

    var body: some View {
        ZStack {
            ForEach(cards) { (card) in
                CardView(card: card)
                .modifier(
                    CardStyleModifier(
                        style: self.style(for: card)
                    )
                )
            }
        }
    }

    private func getIndex(for card: Card) -> Int {
        return cards.firstIndex(where: { $0.id == card.id })!
    }

    private func cardScale(forIndex index: Int, totalCards: Int) -> CGFloat {
        guard totalCards > 1 else { return 1 }

        // Line Ecuation: x = (-y + a) / a
        let scaleFactor: Float = Float(-index + totalCards - 1) / Float(totalCards
```



Preview

100%

```
func getCardView(card: Card) -> ModifiedContent<CardView, CardStyleModifier> {
    CardView(card: card)
    .modifier(
        CardStyleModifier(
            style: self.style(
                for: card,
                frontCardDragOffset: self.frontCardDragOffset
            )
        )
    )
}
```

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 22:15

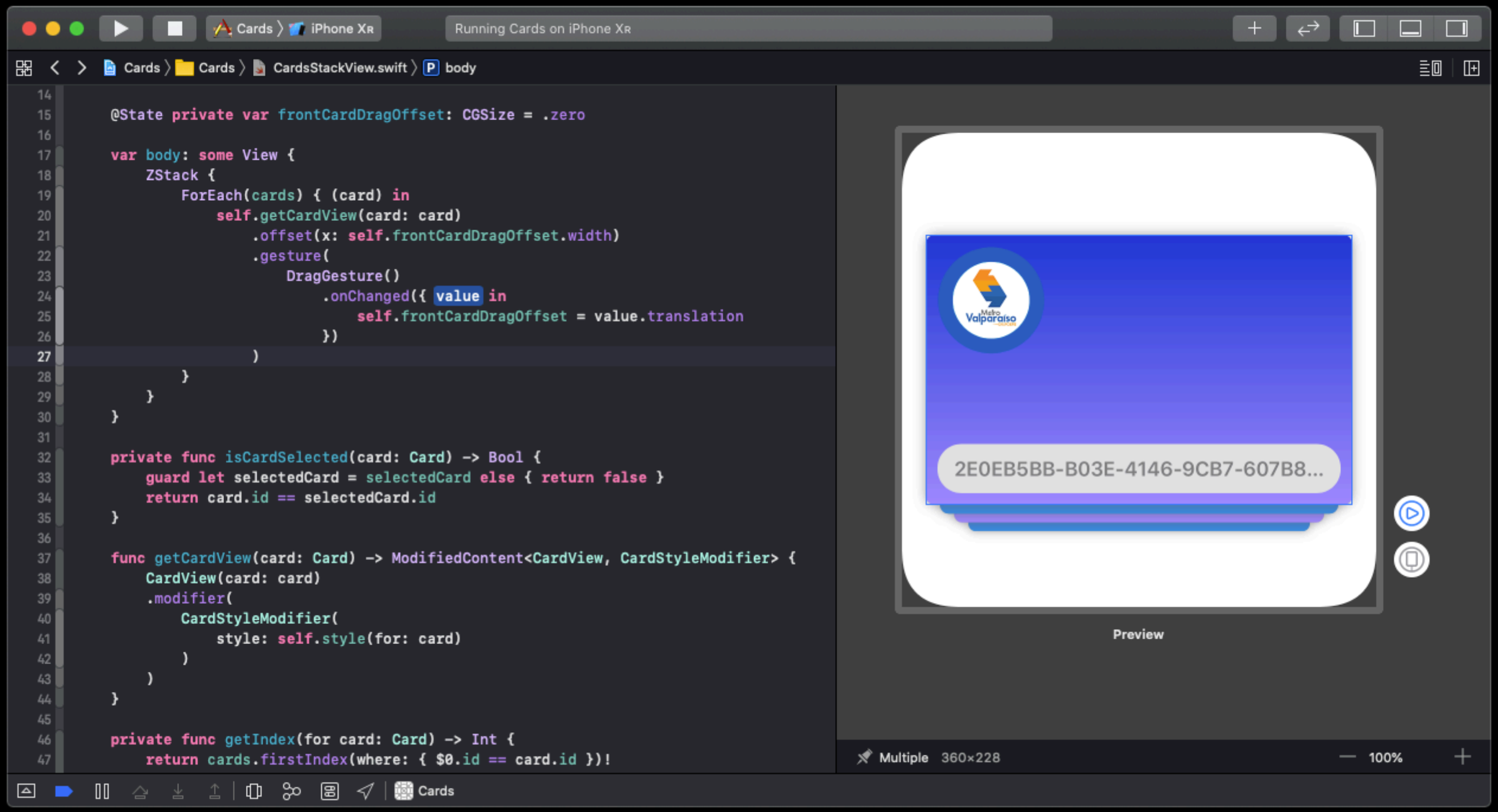
Cards 〉 Cards 〉 CardsStackView.swift 〉 P body

```swift

import SwiftUI

struct CardsStackView: View {

    private struct Constants {
        static let cardsVerticalOffset: Float = 15
        static let cardsScaleOffset: Float = 0.2
    }

    let cards: [Card]

    var body: some View {
        ZStack {
            ForEach(cards) { (card) in
                self.getCardView(card: card)
            }
        }
    }

    func getCardView(card: Card) -> ModifiedContent<CardView, CardStyleModifier> {
        CardView(card: card)
        .modifier(
            CardStyleModifier(
                style: self.style(for: card)
            )
        )
    }

    private func getIndex(for card: Card) -> Int {
        return cards.firstIndex(where: { $0.id == card.id })!
    }

    private func cardScale(forIndex index: Int, totalCards: Int) -> CGFloat {
```

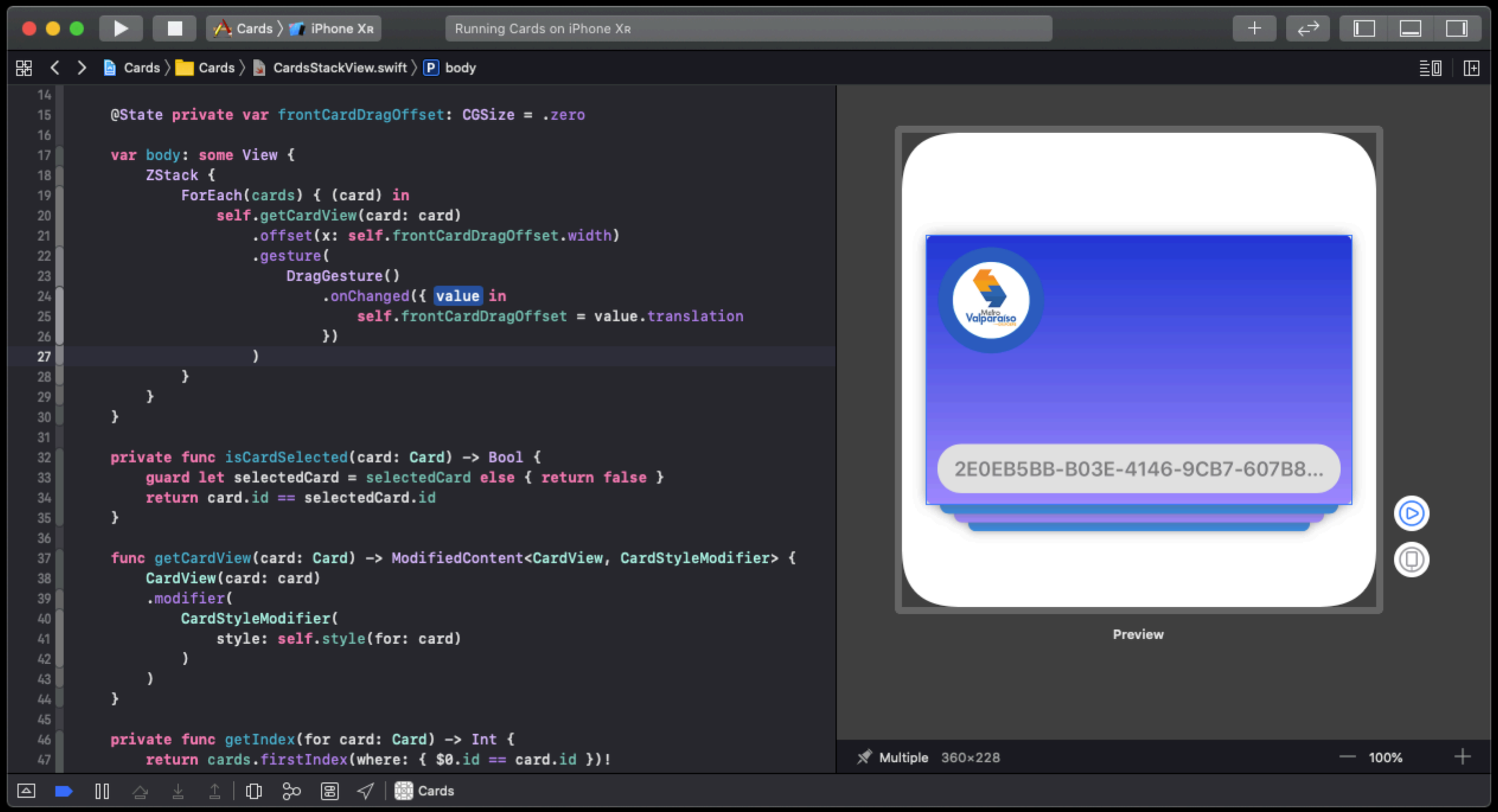
Preview

Multiple 360×228 — 100%

Gestures


Cards 〉 iPhone XR
Running Cards on iPhone XR
Cards 〉 Cards 〉 CardsStackView.swift 〉 P body
14
15 @State private var frontCardDragOffset: CGSize = .zero
16
17 var body: some View {
18 ZStack {
19 ForEach(cards) { (card) in
20 self.getCardView(card: card)
21 .offset(x: self.frontCardDragOffset.width)
22 .gesture(
23 DragGesture()
24 .onChanged({ value in
25 self.frontCardDragOffset = value.translation
26 })
27)
28 }
29 }
30 }
31
32 private func isCardSelected(card: Card) -> Bool {
33 guard let selectedCard = selectedCard else { return false }
34 return card.id == selectedCard.id
35 }
36
37 func getCardView(card: Card) -> ModifiedContent<CardView, CardStyleModifier> {
38 CardView(card: card)
39 .modifier(
40 CardStyleModifier(
41 style: self.style(for: card)
42)
43)
44 }
45
46 private func getIndex(for card: Card) -> Int {
47 return cards.firstIndex(where: { $0.id == card.id })!
Metro Valparaiso
2E0EB5BB-B03E-4146-9CB7-607B8...
Preview
Multiple 360×228
100%
Cards

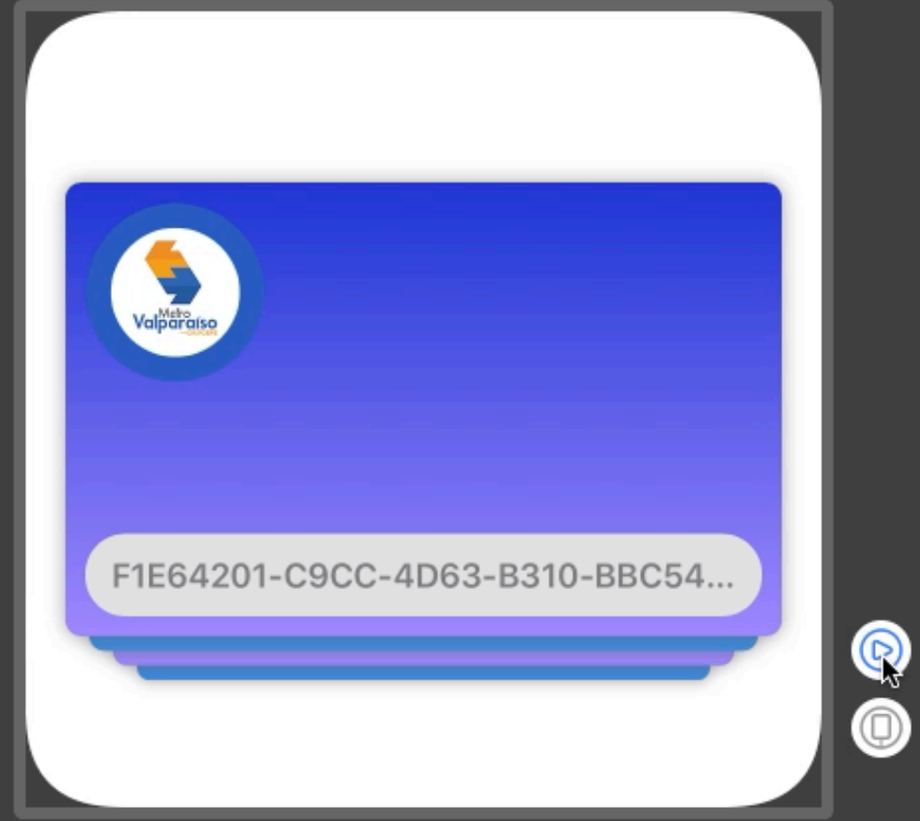
@State


Cards 〉 iPhone XR
Running Cards on iPhone XR
Cards 〉 Cards 〉 CardsStackView.swift 〉 P body
14
15 @State private var frontCardDragOffset: CGSize = .zero
16
17 var body: some View {
18 ZStack {
19 ForEach(cards) { (card) in
20 self.getCardView(card: card)
21 .offset(x: self.frontCardDragOffset.width)
22 .gesture(
23 DragGesture()
24 .onChanged({ value in
25 self.frontCardDragOffset = value.translation
26 })
27)
28 }
29 }
30 }
31
32 private func isCardSelected(card: Card) -> Bool {
33 guard let selectedCard = selectedCard else { return false }
34 return card.id == selectedCard.id
35 }
36
37 func getCardView(card: Card) -> ModifiedContent<CardView, CardStyleModifier> {
38 CardView(card: card)
39 .modifier(
40 CardStyleModifier(
41 style: self.style(for: card)
42)
43)
44 }
45
46 private func getIndex(for card: Card) -> Int {
47 return cards.firstIndex(where: { $0.id == card.id })!
Metro Valparaíso
2E0EB5BB-B03E-4146-9CB7-607B8...
Preview
Multiple 360×228
100%
Cards

Cards 〉 iPhone XR | Running Cards on iPhone XR

Cards 〉 Cards 〉 CardsStackView.swift 〉 P body

```swift
    @State private var frontCardDragOffset: CGSize = .zero

    var body: some View {
        ZStack {
            ForEach(cards) { (card) in
                self.getCardView(card: card)
                    .offset(x: self.frontCardDragOffset.width)
                    .gesture(
                        DragGesture()
                            .onChanged({ value in
                                self.frontCardDragOffset = value.translation
                            })
                    )
            }
        }
    }

    private func isCardSelected(card: Card) -> Bool {
        guard let selectedCard = selectedCard else { return false }
        return card.id == selectedCard.id
    }

    func getCardView(card: Card) -> ModifiedContent<CardView, CardStyleModifier> {
        CardView(card: card)
        .modifier(
            CardStyleModifier(
                style: self.style(for: card)
            )
        )
    }

    private func getIndex(for card: Card) -> Int {
        return cards.firstIndex(where: { $0.id == card.id })!
```



Preview

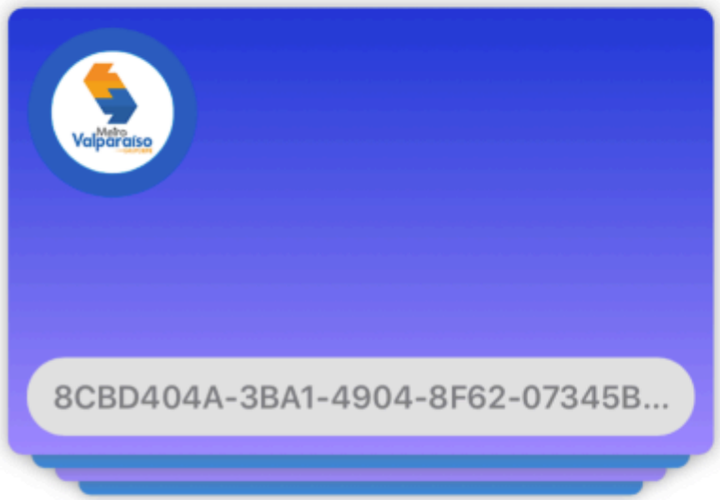
100%

Cards

Cards 〉 iPhone XR | Running Cards on iPhone XR

Cards 〉 Cards 〉 CardsStackView.swift 〉 P body

```swift
    @State private var frontCardDragOffset: CGSize = .zero

    var body: some View {
        ZStack {
            ForEach(cards) { (card) in
                self.getCardView(card: card)
                    .offset(x: self.frontCardDragOffset.width)
                    .gesture(
                        DragGesture()
                            .onChanged({ value in
                                self.frontCardDragOffset = value.translation
                            })
                            .onEnded({ value in
                                self.frontCardDragOffset = .zero
                            })
                    )
            }
        }
    }

    private func isCardSelected(card: Card) -> Bool {
        guard let selectedCard = selectedCard else { return false }
        return card.id == selectedCard.id
    }

    func getCardView(card: Card) -> ModifiedContent<CardView, CardStyleModifier> {
        CardView(card: card)
        .modifier(
            CardStyleModifier(
                style: self.style(for: card)
            )
        )
    }
```



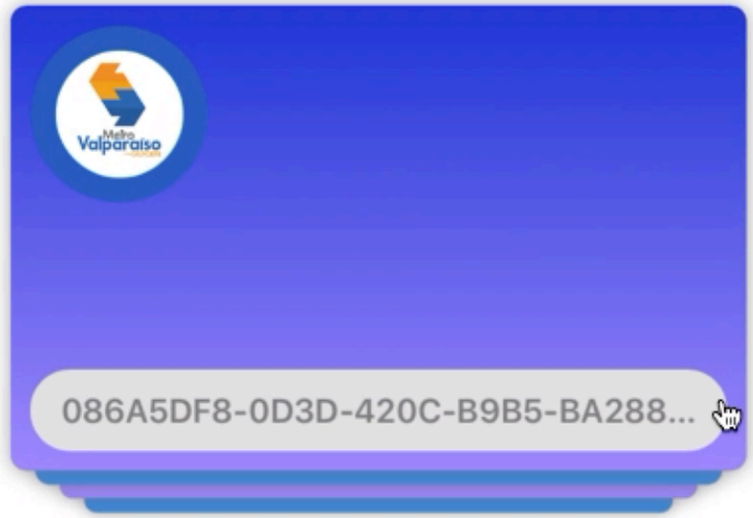
100%

Cards

Cards 〉 iPhone XR | Running Cards on iPhone XR

Cards 〉 Cards 〉 CardsStackView.swift 〉 P body

```swift
@State private var frontCardDragOffset: CGSize = .zero

var body: some View {
    ZStack {
        ForEach(cards) { (card) in
            self.getCardView(card: card)
                .offset(x: self.frontCardDragOffset.width)
                .gesture(
                    DragGesture()
                        .onChanged({ value in
                            self.frontCardDragOffset = value.translation
                        })
                        .onEnded({ value in
                            self.frontCardDragOffset = .zero
                        })
                )
        }
    }
}

private func isCardSelected(card: Card) -> Bool {
    guard let selectedCard = selectedCard else { return false }
    return card.id == selectedCard.id
}

func getCardView(card: Card) -> ModifiedContent<CardView, CardStyleModifier> {
    CardView(card: card)
    .modifier(
        CardStyleModifier(
            style: self.style(for: card)
        )
    )
}
```



100%

Cards

```swift
private func isCardSelected(card: Card) -> Bool {
    guard let selectedCard = selectedCard else { return false }
    return card.id == selectedCard.id
}
```

```swift
func configureTopCard(card: ModifiedContent<CardView, CardStyleModifier>) -> some View  {
    return card
        .gesture(
            DragGesture()
                .onChanged({ value in
                    self.frontCardDragOffset = value.translation
                })
                .onEnded({ value in
                    self.frontCardDragOffset = .zero
                })
        )
}
```

Cards 〉 iPhone XR | Running Cards on iPhone XR | 4

Cards 〉 Cards 〉 CardsStackView.swift 〉 CardsStackView

```
struct CardsStackView: View {

    private struct Constants {
        static let cardsVerticalOffset: Float = 15
        static let cardsScaleOffset: Float = 0.2
    }

    let cards: [Card]
    let selectedCard: Card?
    let didCardMove: () -> Void

    @State private var frontCardDragOffset: CGSize = .zero

    var body: some View {
        ZStack {
            ForEach(cards) { (card) in
                let cardView = self.getCardView(card: card)   Closure containing a declaration...
                if self.isCardSelected(card: card) {   Struct declaration cannot close over valu...
                    self.configureTopCard(card: cardView)   Struct declaration cannot close ov...
                } else {
                    cardView   Struct declaration cannot close over value 'cardView' defined in outer scope
                }
            }
        }
    }

    func configureTopCard(card: ModifiedContent<CardView, CardStyleModifier>) -> some
        View  {
        return card
            .gesture(
                DragGesture()
                    .onChanged({ value in
                        self.frontCardDragOffset = value.translation
                    })
                    .onEnded({ value in
```

Failed to build the scheme "Cards" | Try Again | Diagnostics

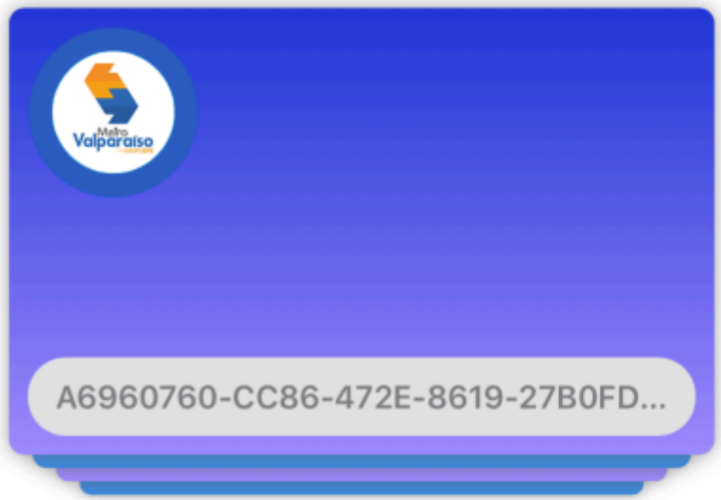
100%

Cards

Cards 〉 iPhone XR | Running Cards on iPhone XR

Cards 〉 Cards 〉 CardsStackView.swift 〉 P body

```
10
11      let cards: [Card]
12      let selectedCard: Card?
13      let didCardMove: () -> Void
14
15      @State private var frontCardDragOffset: CGSize = .zero
16
17      var body: some View {
18          ZStack {
19              ForEach(cards) { (card) -> AnyView in
20                  let cardView = self.getCardView(card: card)
21                  if self.isCardSelected(card: card) {
22                      return AnyView(
23                          self.configureTopCard(card: cardView)
24                              .offset(x: self.frontCardDragOffset.width)
25                      )
26                  } else {
27                      return AnyView(cardView)
28                  }
29              }
30          }
31      }
32
33      func configureTopCard(card: ModifiedContent<CardView, CardStyleModifier>) -> some
            View  {
34          return card
35              .gesture(
36                  DragGesture()
37                      .onChanged({ value in
38                          self.frontCardDragOffset = value.translation
39                      })
40                      .onEnded({ value in
41                          self.frontCardDragOffset = .zero
42                      })
```



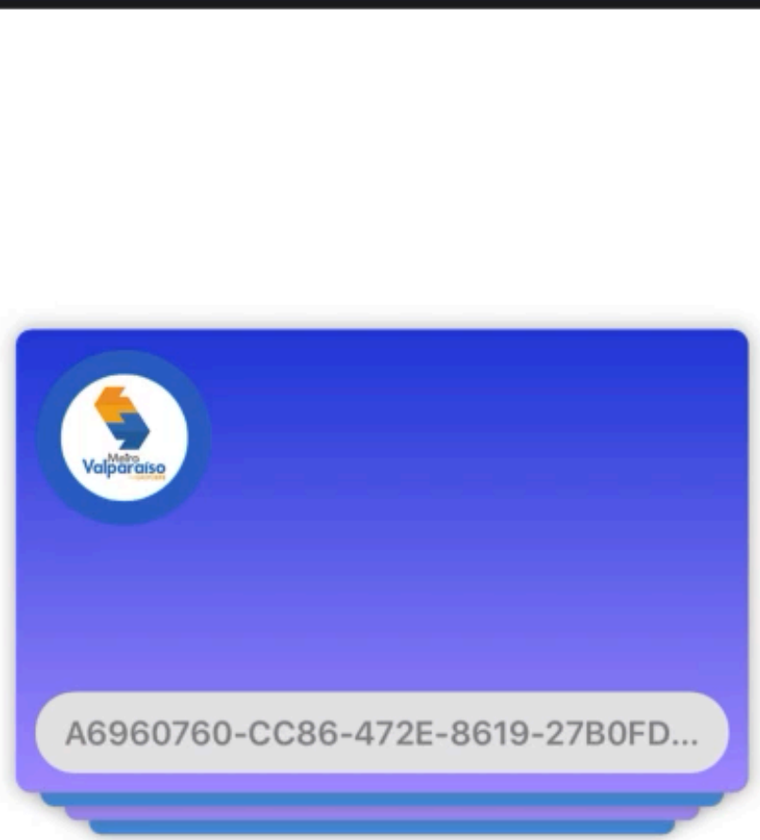
100%

Cards

Cards 〉 iPhone XR | Running Cards on iPhone XR

Cards 〉 Cards 〉 CardsStackView.swift 〉 P body

```
10
11      let cards: [Card]
12      let selectedCard: Card?
13      let didCardMove: () -> Void
14
15      @State private var frontCardDragOffset: CGSize = .zero
16
17      var body: some View {
18          ZStack {
19              ForEach(cards) { (card) -> AnyView in
20                  let cardView = self.getCardView(card: card)
21                  if self.isCardSelected(card: card) {
22                      return AnyView(
23                          self.configureTopCard(card: cardView)
24                              .offset(x: self.frontCardDragOffset.width)
25                      )
26                  } else {
27                      return AnyView(cardView)
28                  }
29              }
30          }
31      }
32
33      func configureTopCard(card: ModifiedContent<CardView, CardStyleModifier>) -> some
            View  {
34          return card
35              .gesture(
36                  DragGesture()
37                      .onChanged({ value in
38                          self.frontCardDragOffset = value.translation
39                      })
40                      .onEnded({ value in
41                          self.frontCardDragOffset = .zero
42                      })
```



100%

Cards

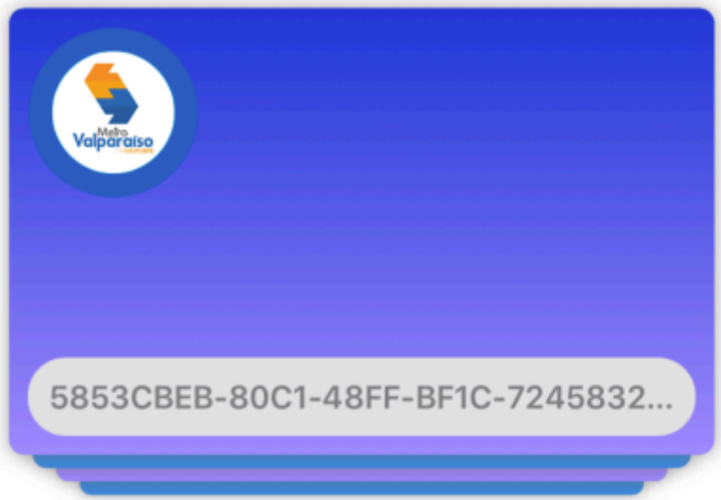
```
func configureTopCard(card: ModifiedContent<CardView, CardStyleModifier>) -> some
    View  {
    return card
        .gesture(
            DragGesture()
                .onChanged({ value in
                    self.frontCardDragOffset = value.translation
                })
                .onEnded({ value in
                    let finalXTranslation = value.translation.width
                    let movesAtEnd = abs(finalXTranslation) >
                        abs(self.frontCardDragOffset.width)

                    self.frontCardDragOffset = .zero
                    if movesAtEnd {
                        self.didCardMove()
                    }
                })
        )
}
```

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 22:40

Cards 〉 Cards 〉 CardsStackView.swift 〉 M configureTopCard(card:)

```
126                             animationPercentage: animationPercentage)
127
128         return CardStyle(scale: scale,
129                          yPos: yPos,
130                          dragOffset: dragOffset,
131                          alpah: alpha)
132     }
133 }
134
135 #if DEBUG
136 struct CardsStackView_Previews : PreviewProvider {
137     static var previews: some View {
138         Group {
139             CardsStackView(cards: Card.previewContent,
140                            selectedCard: Card.previewContent.last,
141                            didCardMove: {})
142                 .scaledToFit()
143         }
144         .padding(20)
145         .previewLayout(.fixed(width: 400, height: 400))
146     }
147 }
148 #endif
149
```

Automatic preview updating paused ⓘ Resume



100%

```swift
private func cardDragOffset(isFrontCard: Bool, frontCardDragOffset: CGSize) ->
    CGFloat {
    guard isFrontCard else { return .zero }
    return frontCardDragOffset.width
}
```

```swift

import SwiftUI

struct CardStyle {
    let scale: CGFloat
    let yPos: CGFloat
    let dragOffset: CGFloat
    let alpah: Double
}

struct CardStyleModifier: ViewModifier {

    private struct Constants {
        static let animation: Animation = .linear(duration: 0.2)
    }

    let style: CardStyle

    func body(content: Content) -> some View {
        content
            .scaleEffect(style.scale)
            .offset(y: style.yPos)
            .offset(x: style.dragOffset)
            .opacity(style.alpah)
            .animation(Constants.animation)
    }
}

```

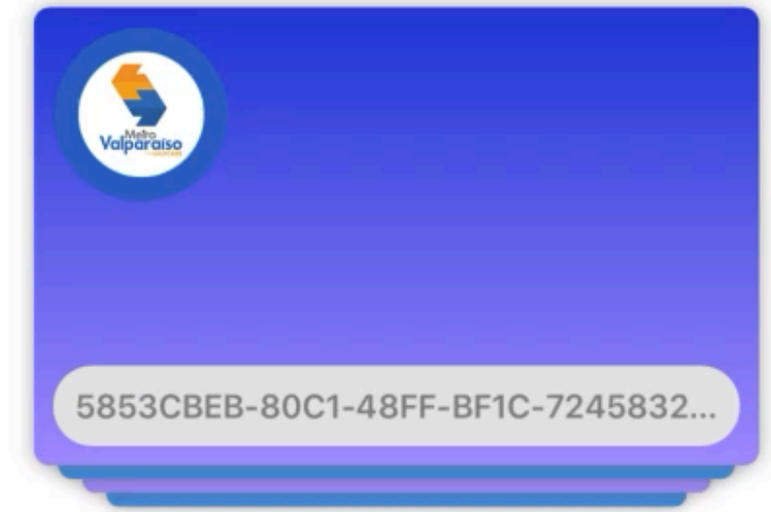
One Eternity Later

Cards 〉 iPhone XR | Cards | Preview Cards: **Succeeded** | Today at 22:40

Cards 〉 Cards 〉 CardsStackView.swift 〉 S CardsStackView

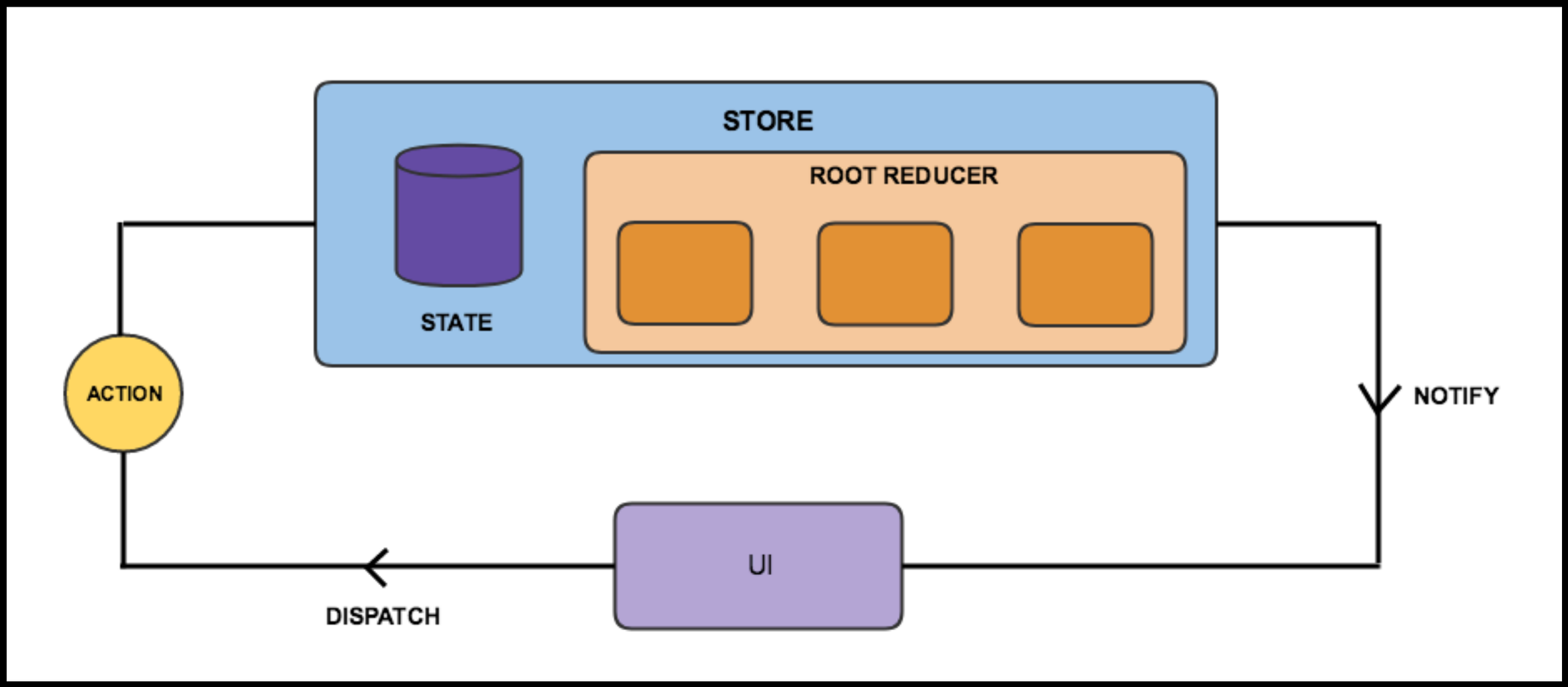
```
            totalCards: totalCards,
            isFrontCard: isFrontCard,
            animationPercentage: animationPercentage
        )

        let yPos = yPossition(
            forIndex: index,
            totalCards: totalCards,
            isFrontCard: isFrontCard,
            animationPercentage: animationPercentage
        )

        let dragOffset = cardDragOffset(isFrontCard: isFrontCard,
                                        frontCardDragOffset: frontCardDragOffset)

        let alpha = cardAlpah(isFrontCard: isFrontCard,
                              animationPercentage: animationPercentage)

        return CardStyle(scale: scale,
                         yPos: yPos,
                         dragOffset: dragOffset,
                         alpah: alpha)
    }
}

#if DEBUG
struct CardsStackView_Previews : PreviewProvider {
    static var previews: some View {
        Group {
            CardsStackView(cards: Card.previewContent,
                           selectedCard: Card.previewContent.last,
                           didCardMove: {})
            .scaledToFit()
        }
```

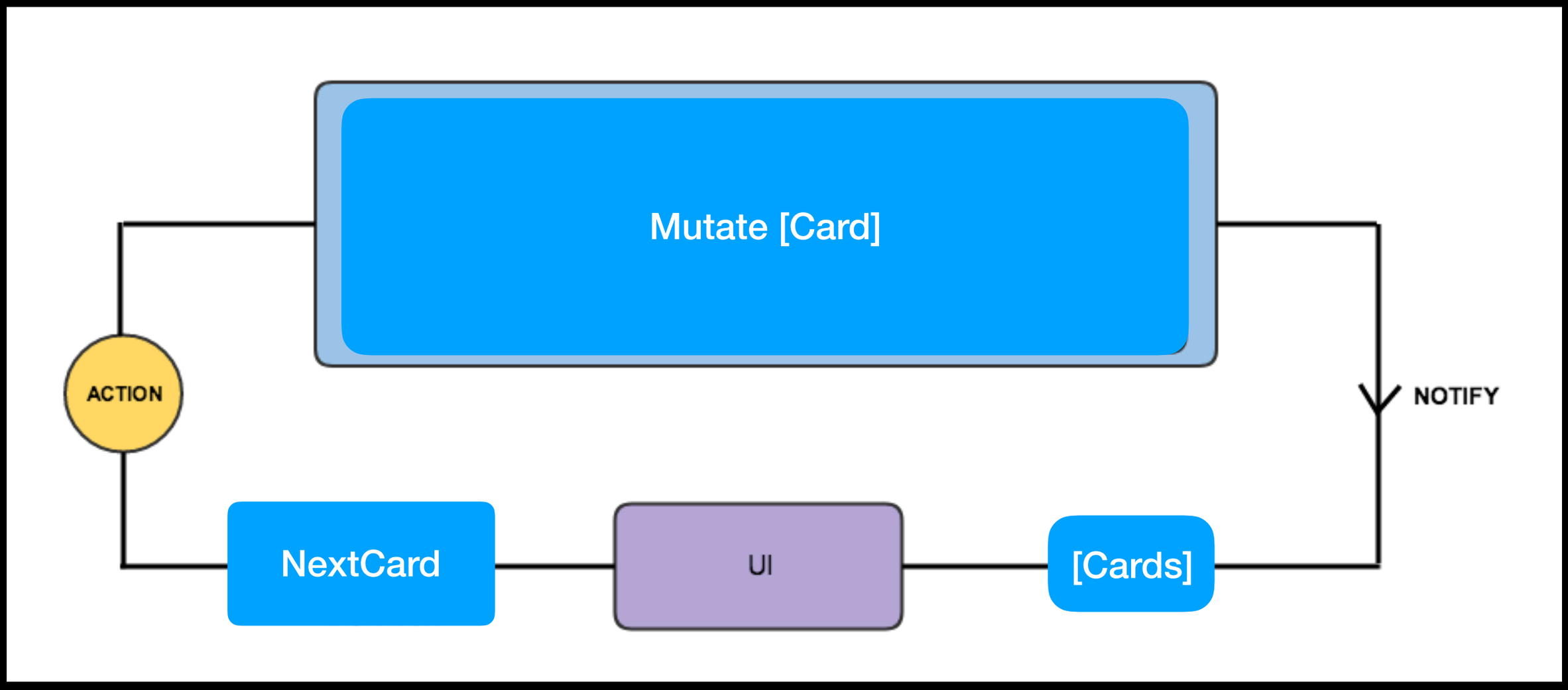
Automatic preview updating paused ⓘ Resume



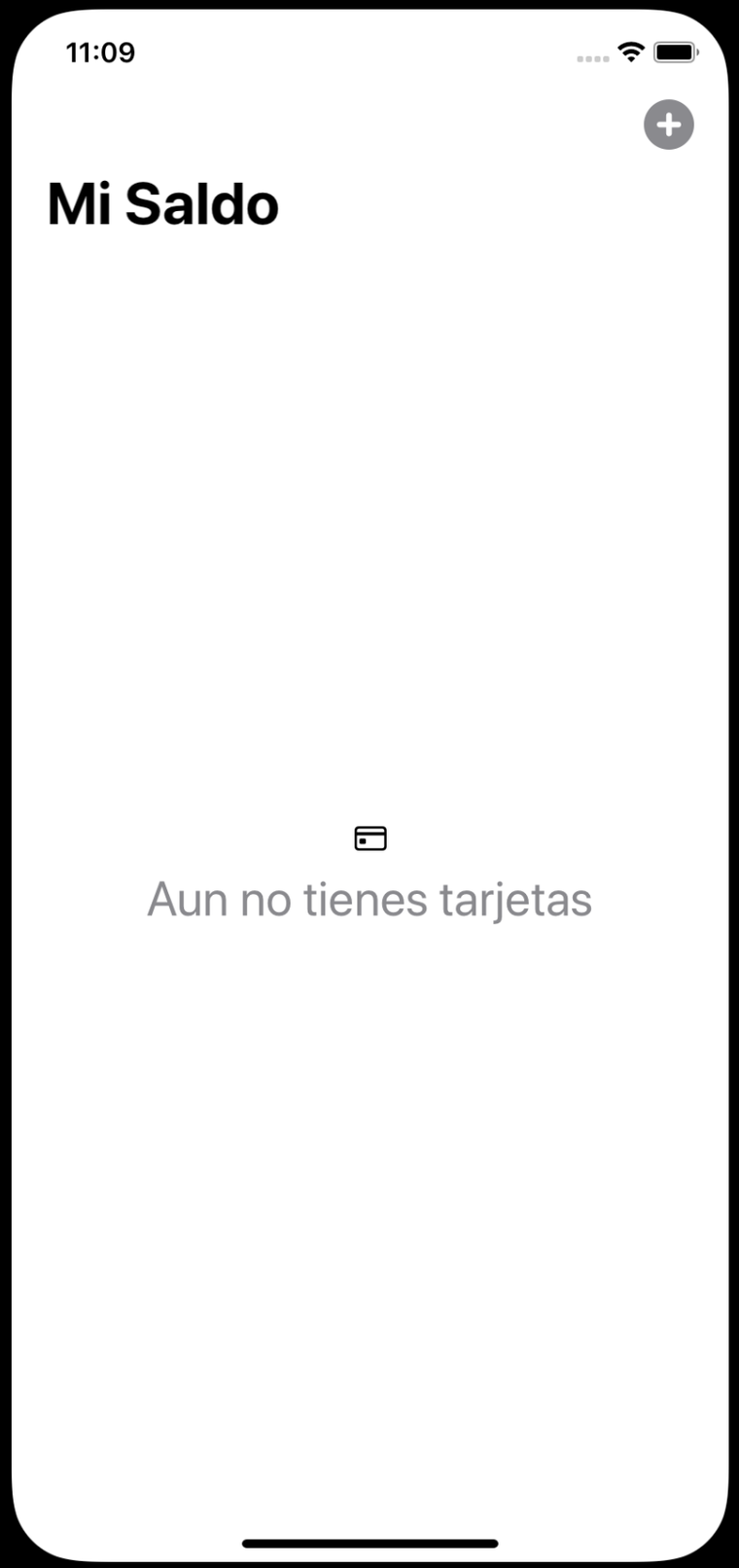
100%


Redux


STORE
ROOT REDUCER
STATE
ACTION
NOTIFY
UI
DISPATCH


Mutate [Card]
ACTION
NOTIFY
NextCard
UI
[Cards]

Empty



Display

```swift
import SwiftUI
import Combine

class HomeStore {
    var state: HomeState
    lazy var actions: HomeActions = HomeActions(delegate: self)

    fileprivate var cards: [Card] = []
    fileprivate var selectedCard: Card?

    init() {
        self.state = HomeState(initialState: .empty)
    }
}
```

ObservableObject

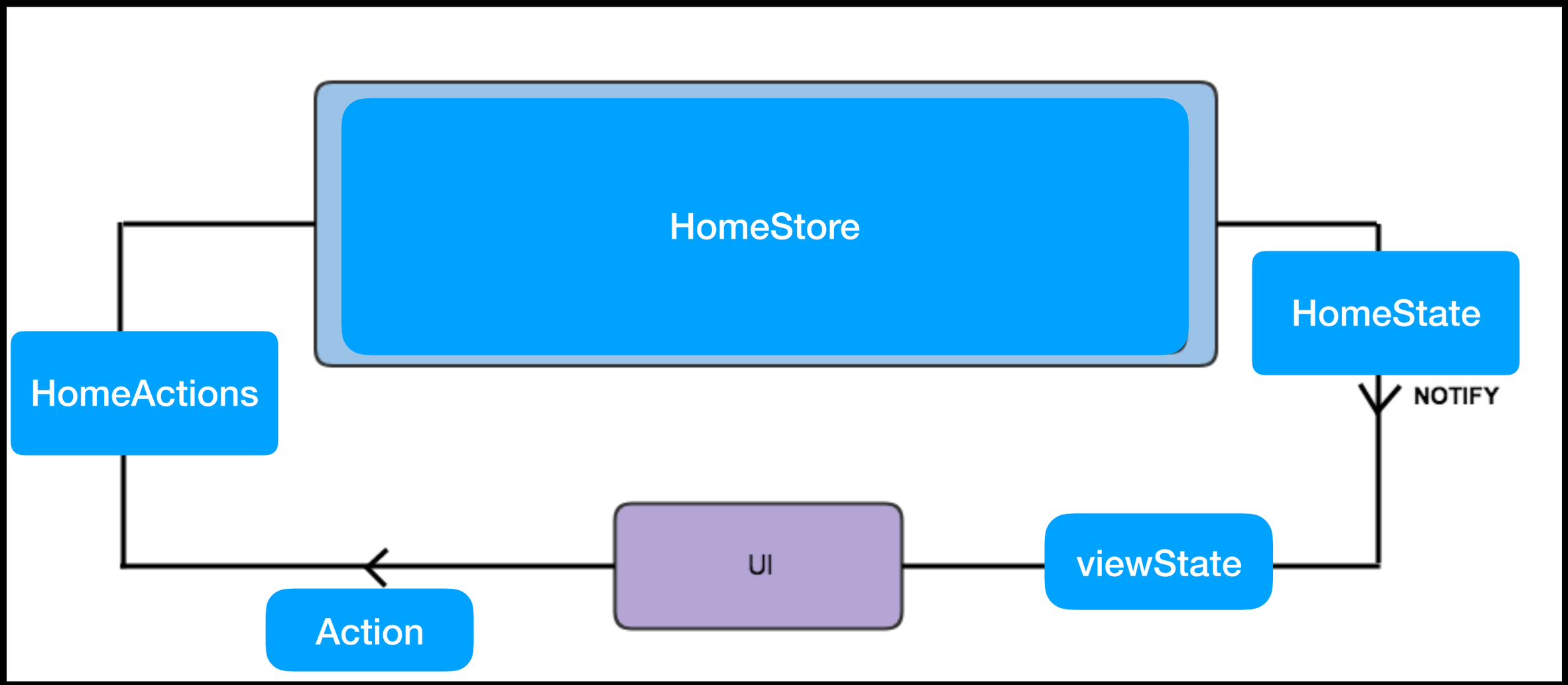
```
import Combine

class HomeState: ObservableObject {
    @Published var viewState: ViewStateCase

    enum ViewStateCase {
        case displaying(cards: [Card], selectedCard: Card?)
        case empty
    }

    init(initialState: HomeState.ViewStateCase) {
        self.viewState = initialState
    }
}
```

@Published

@ObservedObject

```
struct HomeView: View {

    let actions: HomeActions
    @State private var isAddCardPresented = false
    @ObservedObject var state: HomeState

    func currentView(for state: HomeState.ViewStateCase) -> AnyView {
        switch state {
        case .displaying(let cards, let selectedCard):
            return AnyView(
                CardsStackView(
                    cards: cards,
                    selectedCard: selectedCard,
                    didCardMove: {
                     self.actions.moveNextCardToFront()
                })
            )
        case .empty:
            return AnyView(
                CardsEmptyView()
            )
        }
    }

    var body: some View {
        currentView(for: self.$state.viewState.value)
        .onAppear {
            self.actions.loadCards()
        }
    }
}
```

```swift
import Foundation

protocol HomeActionsDelegate: class {
    func loadCards()
    func moveNextCardToFront()
    func addCard(withID id: String, vendor: CardVendor)
}

class HomeActions: HomeActionsDelegate {
    weak var delegate: HomeActionsDelegate?

    init(delegate: HomeActionsDelegate? = nil) {
        self.delegate = delegate
    }

    func loadCards() {
        delegate?.loadCards()
    }
    func moveNextCardToFront() {
        delegate?.moveNextCardToFront()
    }

    func addCard(withID id: String, vendor: CardVendor) {
        delegate?.addCard(withID: id, vendor: vendor)
    }
}
```


HomeStore
HomeState
HomeActions
NOTIFY
UI
viewState
Action

```swift
import UIKit
import SwiftUI

class SceneDelegate: UIResponder, UIWindowSceneDelegate {

    var window: UIWindow?
    lazy var store = HomeStore()

    func scene(_ scene: UIScene, willConnectTo session: UISceneSession, options connectionOptions: UIScene.ConnectionOptions
        guard let windowScene = scene as? UIWindowScene else { return }

        let window = UIWindow(windowScene: windowScene)
        window.rootViewController = UIHostingController(
            rootView: HomeView(actions: store.actions, state: store.state)
        )
        self.window = window
        window.makeKeyAndVisible()
    }
}
```

```swift
#if DEBUG
struct HomeView_Previews : PreviewProvider {
    static var previews: some View {
        Group {
            HomeView(
                actions: HomeActions(),
                state: HomeState(initialState: .empty)
            )

            HomeView(
                actions: HomeActions(),
                state: HomeState(initialState: .displaying(
                    cards: Card.previewContent,
                    selectedCard: Card.previewContent[0]
                    )
                )
            )
        }
    }
}
#endif
```


7:45
Cards
Mi Saldo
bip!
876987696
Saldo
2309
Actualizado
Hace 30 m...
Actualizar
Recarga

Donde continuar

- Proyecto: https://github.com/migueloruiz/MiSaldo-App
- Twitter: Majid Jabrayilov @mecid
- Twitter: Thomas Ricouard @Dimillian
- Twitter: Sarun W. @sarunw
- MovieSwiftUI App: https://github.com/Dimillian/MovieSwiftUI
- SwiftUI By Examples: https://www.hackingwithswift.com/quick-start/swiftui
- About SwiftUI: https://github.com/Juanpe/About-SwiftUI