

SwiftUI를 이용한 ios 앱 개발

메인화면

네비게이션 버튼 클릭시 다음 화면으로
넘어간다.

10:36



네비게이션 버튼

SwiftUI를 이용한 ios 앱 개발

네비게이션 버튼 클릭 시 보이는 화면
배경아이템을 배열로 선언 후 0번부터
4번까지 보여준다.

Binding을 이용하여 메인화면의 그림을
클릭하지 않고 다음화면으로 touch시
활성화 상태가 false, background 색상은
검은색으로 표기된다.

10:37



< Back

배경 아이템 인덱스 0

활성화 상태: false

SwiftUI를 이용한 ios 앱 개발

네비게이션 버튼 클릭 시 보이는 화면
배경아이템을 배열로 선언 후 0번부터
4번까지 보여준다.

Binding을 이용하여 메인화면의 그림을
클릭하지 않고 다음화면으로 touch시
활성화 상태가 false, background 색상은
검은색으로 표기된다.

10:37



< Back

배경 아이템 인덱스 1

활성화 상태: false

SwiftUI를 이용한 ios 앱 개발

배경아이템 인덱스 2번 화면

10:37

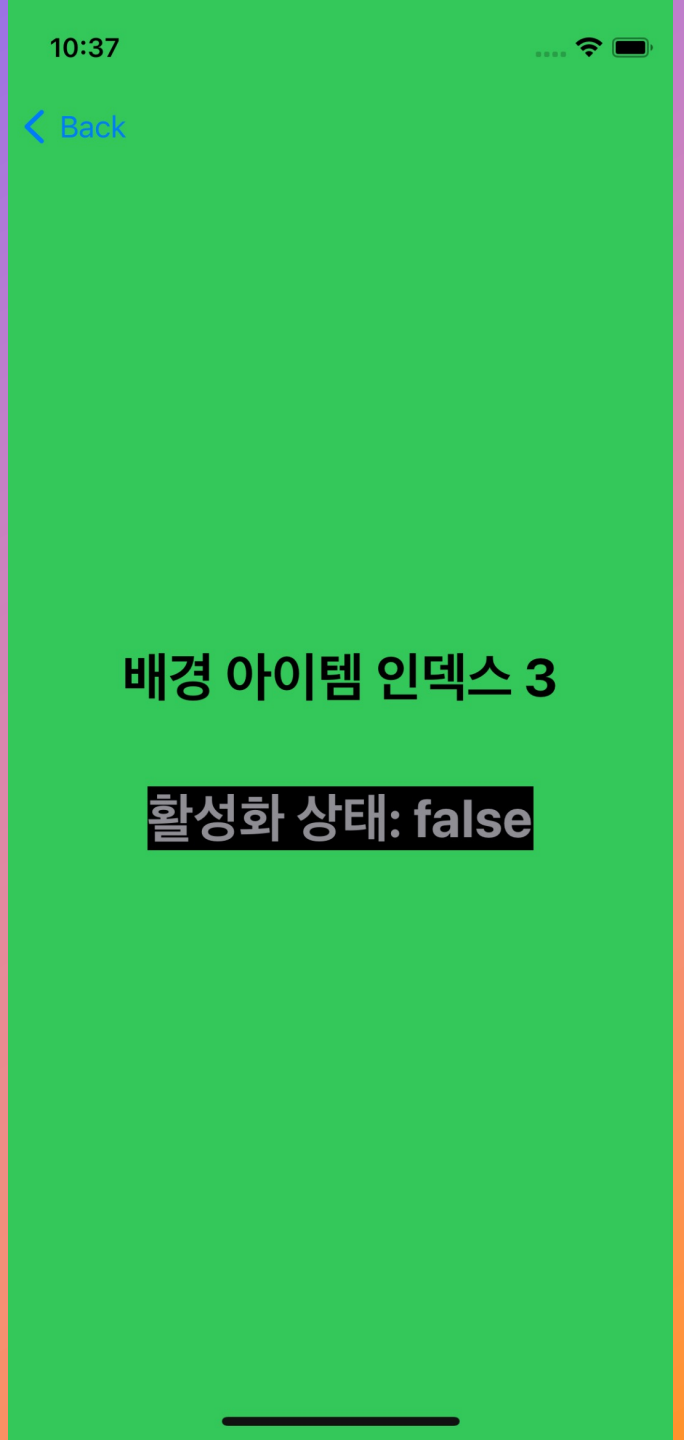


배경 아이템 인덱스 2

활성화 상태: false

SwiftUI를 이용한 ios 앱 개발

배경아이템 인덱스 3번 화면



SwiftUI를 이용한 ios 앱 개발

배경아이템 인덱스 4번 화면

10:37



< Back

배경 아이템 인덱스 4

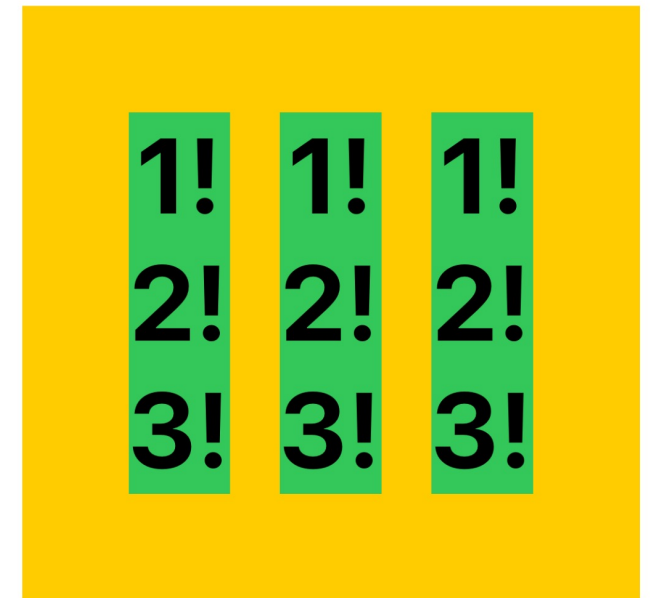
활성화 상태: false

SwiftUI를 이용한 ios 앱 개발

메인화면

그림 클릭 후 네비게이션 버튼 클릭시
다음 화면으로 넘어간다.

10:37



네비게이션 버튼

SwiftUI를 이용한 ios 앱 개발

네비게이션 버튼 클릭 시 보이는 화면
배경아이템을 배열로 선언 후 0번부터
4번까지 보여준다.

Binding을 이용하여 메인화면의 그림을
클릭한 후 다음화면으로 touch시 활성화
상태가 true, background 색상은
노란색으로 표기된다.

10:37



< Back

배경 아이템 인덱스 0

활성화 상태: true

코드리뷰

myVStackView

데이터를 연동시키기 위해 Binding 사용.
isActivated를 Bool타입으로 선언 후
제네릭을 이용하여 결과가 false였을시
백그라운드 컬러는 green으로, true였을 시
Red로 주었다.

○

```
2 // myVstack.swift
3 // swiftUI_tutorial_#2
4 //
5 // Created by 정혜윤 on 2022/01/01.
6 //
7
8 import SwiftUI
9
10 struct myVstackView : View{
11
12     // 데이터를 연동시킨다
13     @Binding
14     var isActivated : Bool
15
16     // 생성자
17     init(isActivated: Binding<Bool> = .constant(false)){ _isActivated = isActivated
18     }
19
20
21     var body: some View{
22         VStack{
23             Text("1!")
24                 .fontWeight(.bold)
25                 .font(.system(size:60))
26             Text("2!")
27                 .fontWeight(.bold)
28                 .font(.system(size:60))
29             Text("3!")|
30                 .fontWeight(.bold)
31                 .font(.system(size:60))
32         }
33         .background(self.isActivated ? Color.green : Color.red)
34         .padding(self.isActivated ? 10 : 0)
35     }
```

코드리뷰

ContentView에서는 아래 함수 이용

//탭 재스처 추가

.onTapGesture {

+ print("HStack 이 클릭되었다")

- //애니메이션과 함께
withAnimation

{ //toggle() true이면 false로
false이면 true

self.isActivated.toggle() }

}//Hstack

결과가 변할 수 있기 때문에

@State 이용

```
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11     @State
12     private var isActivated:Bool = false
13     var body: some View {
14
15         NavigationView{
16             VStack{
17                 HStack{
18                     myVstackView(isActivated: $isActivated)
19                     myVstackView(isActivated: $isActivated)
20                     myVstackView(isActivated: $isActivated)
21
22                 }
23
24                 .padding(isActivated ? 50.0 : 10.0)
25
26                 .background(isActivated ? Color.yellow : Color.black)
27
28
29                 //탭 재스처 추가
30                 .onTapGesture {
31                     print("HStack 이 클릭되었다.")
32
33                     //애니메이션과 함께
34                     withAnimation{ //toggle() true이면 false로 false이면 true
35                         self.isActivated.toggle() }
36                 }//HStack
37
38
39                 //네비게이션 버튼(링크)
```

코드리뷰

MyTextView

@Binding 이용하여 데이터 연동
배경색은 배열을 사용

+

●

○

```
1 //
2 // MyTextView.swift
3 // swiftUI_tutorial_#2
4 //
5 // Created by 정혜윤 on 2022/01/01.
6 //
7
8 import SwiftUI
9
10 struct MyTextView: View{
11
12     @Binding
13     var isActivated : Bool
14
15     // 생성자
16     init(isActivated: Binding<Bool> = .constant(false)){ _isActivated = isActivated
17     }
18
19
20
21
22     @State //변화를 감지->view에 적용
23     private var index:Int = 0
24
25     //배경색 배열 준비
26
27     private let backgroundColors = [
28         Color.red,
29         Color.yellow,
30         Color.blue,
31         Color.green,
32         Color.orange
33     ]
34
```

코드리뷰

MyTextView

If else문 사용

배경 아이템이 전부 끝났을 시
(인덱스는 0부터 시작하므로
-1이 되면 배경아이템을 전부 보여줬다는
뜻)

Index를 다시 0으로 되돌린다

Else문 이용하여

인덱스가 0이 되었을 시 처음으로
되돌아가도록 함

```
34
35     var body: some View{
36         VStack{
37
38             Spacer()
39
40             Text("배경 아이템 인덱스 \(self.index)")
41                 .font(.system(size: 30))
42                 .fontWeight(.bold)
43                 .frame(minWidth:0,maxWidth: .infinity,
44                     minHeight:0,maxHeight: 100)
45
46             Text("활성화 상태: \(String(isActivated))")
47                 .font(.system(size: 30))
48                 .fontWeight(.bold)
49                 .foregroundColor(self.isActivated ? Color.yellow : Color.gray)
50                 .background(Color.black)
51
52             Spacer()
53
54         }.background(backgroundColors[index])
55         .edgesIgnoringSafeArea(.all)
56         .onTapGesture {
57             print("배경 아이템이 클릭 되었다.")
58
59             if(self.index == self.backgroundColors.count - 1 ){
60                 self.index=0
61             }else{
62                 self.index += 1
63             }
64         }
65     }
66 }
67
```