SwiftUI와 UIKit 통합하기

- PDF 파일 표시하기
- 웹사이트 표시하기
- SwiftUI를 스토리보드 프로젝트에 통합하기

SwiftUI와 UIKit 통합 요약

- SwiftUI는 다양한 화면 크기에 적응할 수 있고 UIKit보다 훨씬 적은 코드로 사용자 인터페이스를 만들 수 있지만, UIKit에 있는 많은 기능이 아직 SwiftUI에서 사용할 수 없음
- SwiftUI에 더 많은 기능이 추가될 때까지 기다리거나, SwiftUI와 UIKit을 결합하여 두 프레임워크의 장점을 모두 활용할 수 있음
- UIKit 뷰를 SwiftUI 프로젝트에 통합하려면 UIKit을 가져오고, UIViewRepresentable 프로토콜을 사용하여 UIKit 뷰를 감싸는 구조체를 만들어야 함
- 구조체 내부에는 특정 UIView를 생성하는 makeUIView 함수와 해당 UIView를 업데이트하는 updateUIView 함수가 필요함
- 마지막으로 SwiftUI 프로젝트에서 UIKit 뷰를 표시하려면 구조체 이름을 참조하면 됨

PDF 파일 표시하기

- SwiftUI는 아직 PDFKit을 지원하지 않지만, UIViewRepresentable 구조체 내부에 PDFKit을 래핑하여 사용할 수 있음
- PDF 파일을 표시하려면 PDFKit을 가져오고, UIViewRepresentable을 준수하는 구조체를 생성해야 함
- 구조체 내부에는 PDF 파일 이름과 위치를 지정하는 URL 상수가 필요함
- makeUIView 함수는 PDFView를 반환하고, updateUIView 함수는 PDFView를 업데이트함
- 마지막으로 ContentView에서 구조체 이름을 호출하고 PDF 파일의 URL을 전달하면 됨

웹사이트 표시하기

- WebKit은 SwiftUI에서 사용할 수 없는 또 다른 Apple 프레임워크이지만, UIViewRepresentable 구조체 내부에 래핑하여 사용할 수 있음
- 웹사이트를 표시하려면 WebKit을 가져오고, UIViewRepresentable을 준수하는 구조체를 생성해야 함
- 구조체 내부에는 웹사이트의 URL 주소를 저장할 URL 상수가 필요함
- makeUIView 함수는 WKWebView를 반환하고, updateUIView 함수는 웹사이트를 로드함
- 마지막으로 ContentView에서 구조체 이름을 호출하고 웹사이트 주소를 문자열로 전달하면 됨

SwiftUI를 스토리보드 프로젝트에 통합하기

- 스토리보드로 생성된 기존 프로젝트를 수정하고 점진적으로 SwiftUI 기능을 추가해야 할 수 있음
- 스토리보드 프로젝트에 SwiftUI 파일을 통합하려면 다음 단계를 따라야 함:
  - 스토리보드 프로젝트 생성
  - 뷰 컨트롤러를 내비게이션 컨트롤러에 임베드
  - 뷰 컨트롤러에 버튼 추가
  - Hosting View Controller를 스토리보드에 추가하고 버튼과 연결
  - SwiftUI 뷰 파일 생성
  - 세그 액션 함수를 사용하여 SwiftUI 뷰를 로드하고 데이터 전달
- 이렇게 하면 스토리보드로 빠르고 쉽게 사용자 인터페이스를 만들면서 SwiftUI의 기능을 활용할 수 있음

ContentView.swift (PDF 파일 표시)

이 파일은 PDFKit 프레임워크를 사용하여 PDF 파일을 표시하는 SwiftUI 뷰를 정의합니다. UIViewRepresentable 프로토콜을 준수하는 ViewMe 구조체를 생성하고, PDF 파일의 URL을 전달받아 PDFView를 생성하고 업데이트합니다. ContentView에서는 ViewMe 구조체를 호출하고 PDF 파일의 URL을 전달하여 PDF 파일을 표시합니다.

ContentView.swift (PDF 파일 표시)

```swift
import SwiftUI
import PDFKit

struct ViewMe: UIViewRepresentable {
    let url: URL

    func makeUIView(context: UIViewRepresentableContext<ViewMe>) -> PDFView {
        let pdfView = PDFView()
        pdfView.document = PDFDocument(url: url)
        return pdfView
    }

    func updateUIView(_ uiView: PDFView, context: UIViewRepresentableContext<ViewMe>) {
    }
}

struct ContentView: View {
    let fileURL = Bundle.main.url(forResource: "ds11", withExtension: "pdf")

    var body: some View {
        ViewMe(url: fileURL!)
    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

ContentView.swift (웹사이트 표시)

이 파일은 WebKit 프레임워크를 사용하여 웹사이트를 표시하는 SwiftUI 뷰를 정의합니다. UIViewRepresentable 프로토콜을 준수하는 WebView 구조체를 생성하고, 웹사이트의 URL을 전달받아 WKWebView를 생성하고 웹사이트를 로드합니다. ContentView에서는 WebView 구조체를 호출하고 웹사이트 주소를 문자열로 전달하여 웹사이트를 표시합니다.

ContentView.swift (웹사이트 표시)

```swift
import SwiftUI
import WebKit

struct WebView: UIViewRepresentable {
    let url: URL

    func makeUIView(context: UIViewRepresentableContext<WebView>) -> WKWebView {
        return WKWebView()
    }

    func updateUIView(_ webView: WKWebView, context: UIViewRepresentableContext<WebView>) {
        let request = URLRequest(url: url)
        webView.load(request)
    }
}

struct ContentView: View {
    var body: some View {
        WebView(url: URL(string: "https://www.apple.com")!)
    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

ViewController.swift (SwiftUI를 스토리보드 프로젝트에 통합)

이 파일은 스토리보드 기반 프로젝트에서 SwiftUI 뷰를 호스팅하는 뷰 컨트롤러입니다. 스토리보드에서 Hosting View Controller를 추가하고 버튼과 연결한 후, 세그 액션 함수를 사용하여 SwiftUI 뷰를 로드하고 데이터를 전달합니다.

ViewController.swift (SwiftUI를 스토리보드 프로젝트에 통합)

```
import UIKit
import SwiftUI

class ViewController: UIViewController {

    @IBSegueAction func openSwiftUIView(_ coder: NSCoder) -> UIViewController? {
        return UIHostingController(coder: coder, rootView: SwiftUIView(name: "Nancy"))
    }
}
```

요약

SwiftUI는 미래의 사용자 인터페이스 개발 방식이지만, 아직 스토리보드에서 제공하는 많은 기능이 부족하다. 다행히도 UIKit, PDFKit, WebKit 등 다른 프레임워크의 기능을 SwiftUI 프로젝트에 결합할 수 있다. 또한 기존 스토리보드 기반 프로젝트에 SwiftUI를 추가할 수도 있다. 이렇게 함으로써 Xcode는 SwiftUI와 스토리보드의 장점을 모두 활용할 수 있게 해준다.

END