14 장. 비디오 재생 앱 만들기

아이폰 앱에서 비디오 파일을 재생하는 방법에 대해 알아본다.

애플 iOS 에서 제공하는 AVPlayerViewController 를 사용하면 앱내부에 저장되어 있는 비디오 파일뿐만 아니라 외부에 링크된 비디오 파일도 간단하게 재생할 수 있다.

14-1 비디오 재생 앱이란

비디오 플레이어는 아이폰 사용자들이 가장 많이 사용하는 앱 중의 하나이다. 등하굣길이나 출퇴근길에 영화를 감상하거나 동영상 강좌를 듣는 사람들을 쉽게 볼 수 있다. 아이폰에서의 비디오 재생 방법을 잘 활용하면 다음과 같은 비디오 재생 앱도 만들 수 있다.

AVPlayer 앱

HPlayer 앱

14-2 비디오 재생 앱을 위한기본 환경 구성하기

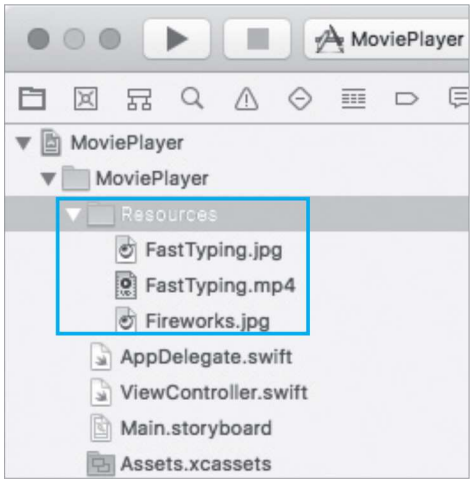
1. Xcode 를 실행한 후 'MoviePlayer'라는 이름으로 프로젝트를 만든다.

2. 뷰 컨트롤러 크기 조절

아이폰 모양의 뷰 컨트롤러 크기를 상황에 맞게 조절한다.

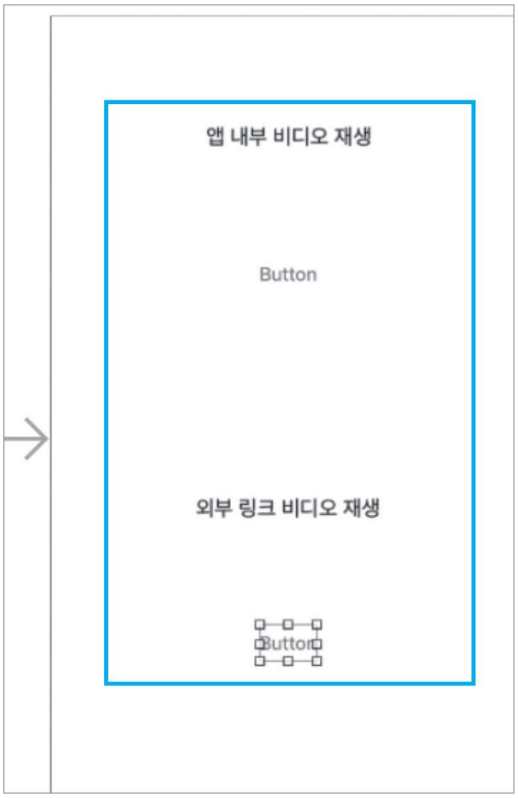
3. 미디어 파일을 추가한다.

비디오 재생 앱에 필요한 미디어 파일(비디오 파일 한 개, 이미지 파일 두개)을 추가해 보자. 왼쪽의 내비게이터 영역에 'Resources'라는 이름의 새 폴더를 만들고, 그안에 파일을 드래그 앤 드롭하여 추가한다. 추가한 미디오 파일들을 모두 선택한 후 오른쪽 인스펙터 영역에서[Target Membership]의 [MoviePlyer]에 체크를 한다.

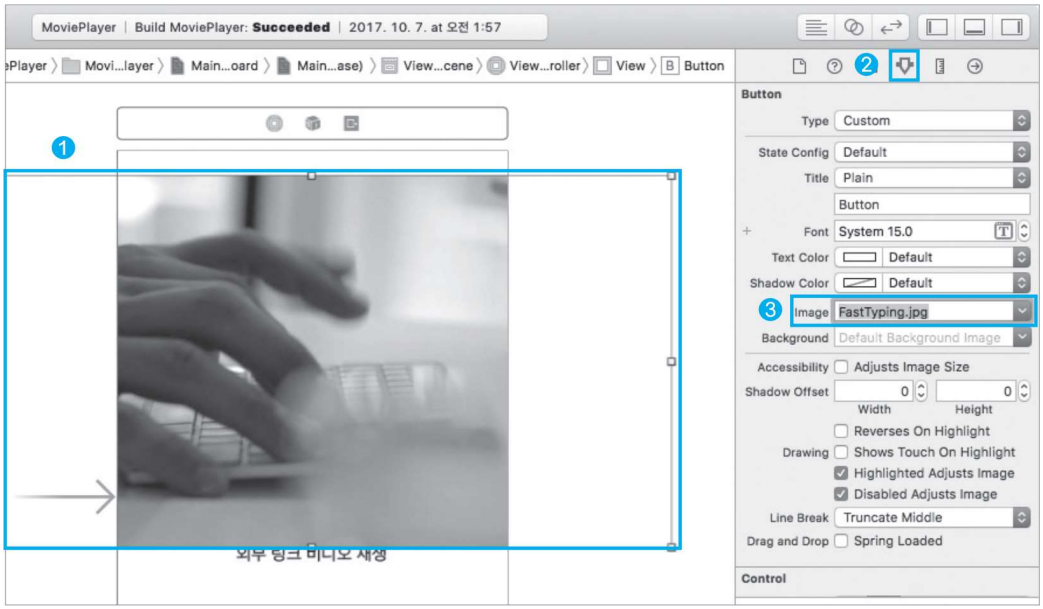


4. 스토리보드 꾸미기

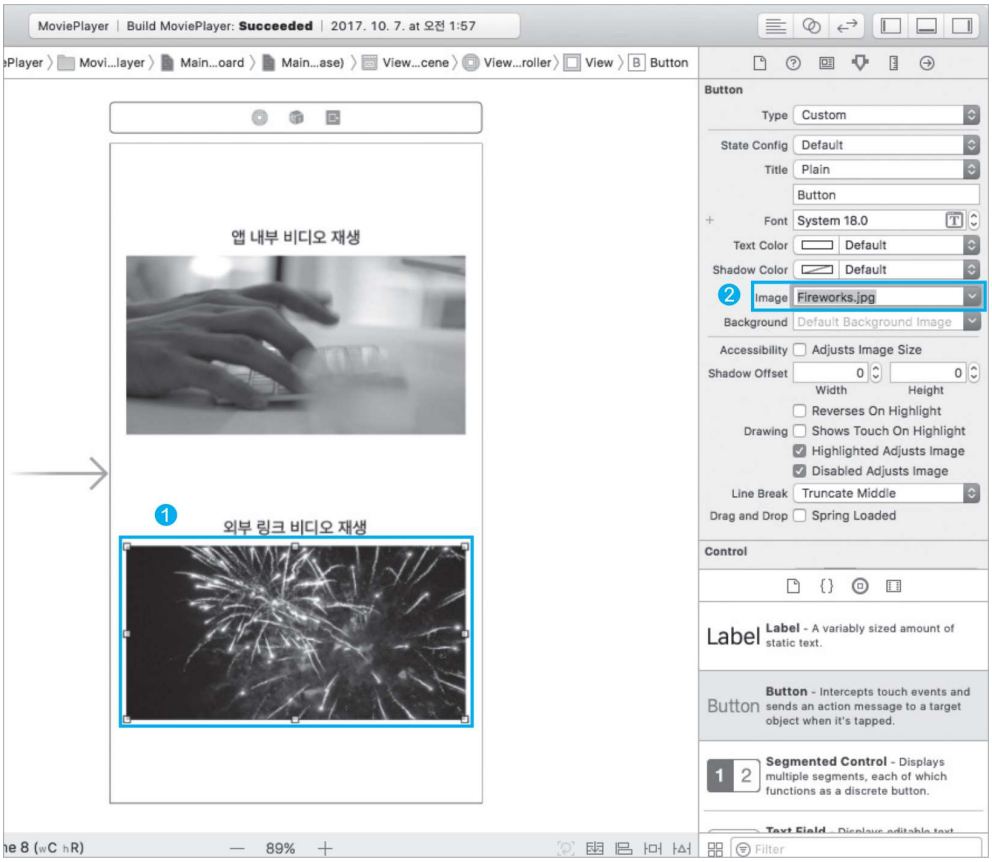
오른쪽 아랫부분의 오브젝트 라이브러리에서 [레이블(Label)]과 [버튼(Button)]을 찾아 스토리보드로 끌어와 오른쪽 그림처럼 배치한 후, 레이블 내용을 수정한다.



5. 이제 첫 번째 버튼에 이미지를 나타내 본다. 첫 번째 버튼을 클릭한 후 오른쪽의 인스펙터 영역에서 [Atrribute inspector]버튼을 클릭하고, Image 에서 [FastTyping.jpg]를 선택한다. 그리고 이미지를 적절히 조절한다.



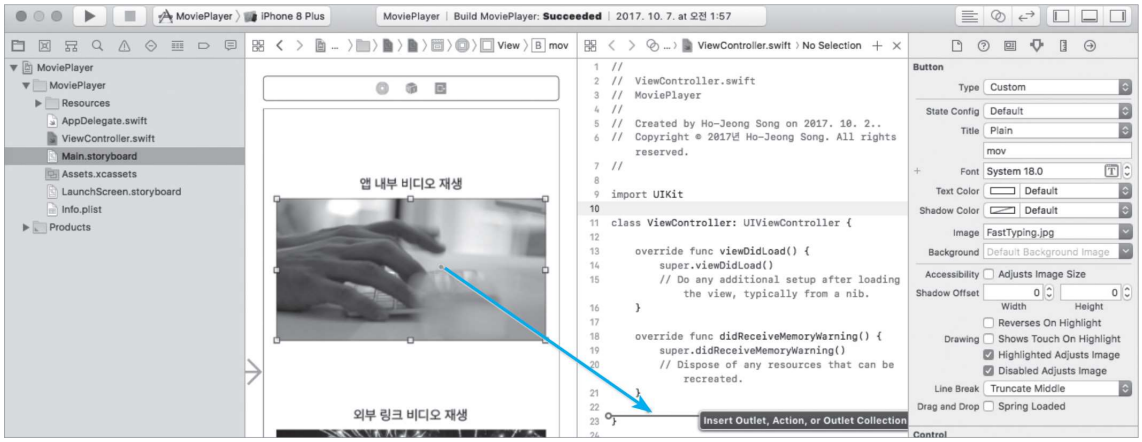
6. 같은 방법으로 두 번째 버튼의 이미지를 [Fireworks.jpg]로 선택하여 추가하고 이미지 크기를 조절한다,

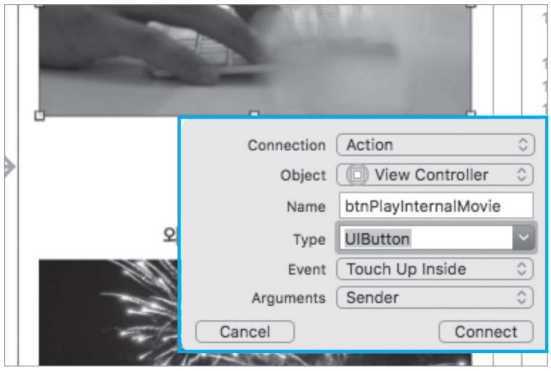


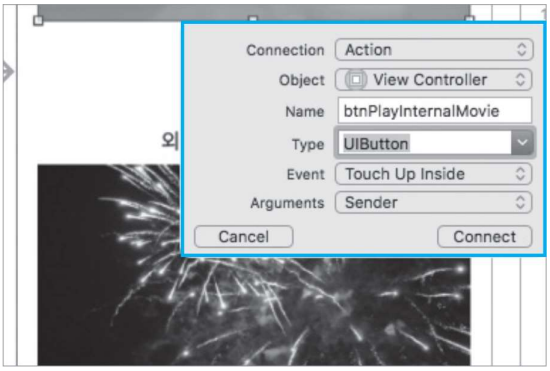
14-3 액션 함수 추가하기

1. 액션 함수를 추가하기 위해 오른쪽 윗부분의 [Show the Assistant editor]버튼을 클릭하여 보조편집기 영역을 연다.

2. 이미지로 만든 첫번째 버튼을 마우스 오른쪽 버튼으로 클릭하여 보조 편집기 영역에 끌어다 놓는다. 자세한 위치와 설정은 아래 표를 참고한다. 같은 방법으로 두 번째 버튼의 액션 함수도 추가한다.







위치	소스의 가장 아래쪽 '}' 바로 위
연결(Connection)	Action
이름(Name)	- 첫 번째 버튼: btnPlayInternalMovie - 두 번째 버튼: btnPlayExternalMovie
유형(Type)	UIButton

3. 액션 함수가 모두 추가되었다.

```
@IBAction func btnPlayInternalMovie(_ sender: UIButton) {
}
@IBAction func btnPlayExternalMovie(_ sender: UIButton) {
}
```

14-4 비디오 재생 앱 기능 구현하기

이제 두 개의 버튼에 각각 '내부 비디오'와 '외부 링크 비디오'를 재생하는 코드를 작성해 보자. 그리고 두 코드에서 동일하게 입력한 부분을 새로운 함수로 정의한다.

1. 화면 모드 수정 및 헤더 파일 추가하기

코딩을 위해 화면 모드를 스텐더드 애디터 모드로 수정하고 비디오 관련 헤더 파일을 추가한다.

```
1  import UIKit
2  import AVKit
3
4  class ViewController: UIViewController {
5
6  ...
7  }
```

2행에 있는 "import AVKit" 를 추가한다

2. '앱 내부 비디오 재생" 코드 작성하기

먼저 앱 내부에 복사해 놓은 비디오를 재생하는 코드를 작성한다. 앞에서 추가한 액션 함수 btnPlayIntenalMovie 안에 다음코드를 입력한다.

```
1  @IBAction func btnPlayInternalMovie(_ sender: UIButton) {
2          // 내부 파일 mp4
3          let filePath:String? = Bundle.main.path(forResource:
4                                  "FastTyping", ofType: "mp4")
5          let url = NSURL(fileURLWithPath: filePath!)
6
7          let playerController = AVPlayerViewController()
8
9          let player = AVPlayer(url: url as URL)
10         playerController.player = player
11
12         self.present(playerController, animated: true) {
13             player.play()
14         }}
15
```

3~4 행은 우선 비디오 파일명을 사용하여 비디오가 저장된 앱 내부의 파일 경로를 받아온다.

5 행은 앱 내부의 파일명을 NSURL 형식으로 변경한다.

6 행은 AVPlayerViewController 의 인스턴스를 생성한다,

10 행은 앞에서 얻은 비디오 URL 로 초기화된 AVPlayer 의 인스턴스를 생성한다.

11 행은 AVPlayerViewController 의 player 속성에 10 행에서 생성한 AVPlayer 인스턴스를

할당한다.
13 행은 비디오를 재생한다.

3. '외부 링크 비디오 재생"코드 작성하기
외부에 링크된 비디오를 재생하는 코드를 작성한다. 외부에 링크된 비디오를 재생하는 방법은 앞의 앱 내부 비디오를 재생하는 방법에서 url 을 얻는 방법만 다르고 그 외의 비디오를 재생하는 방법은 모두 동일하다. NSURL 형식의 url 을 얻는 방법은 다음과 같다.

```
@IBAction func btnPlayExternalMovie(_ sender: UIButton) {
        // 외부 파일 mp4
        let url = NSURL(string:
"https://dl.dropboxusercontent.com/s/e38auz050w2mvud/Fireworks.mp4")!

      let playerController = AVPlayerViewController()

        let player = AVPlayer(url: url as URL)
        playerController.player = player

        self.present(playerController, animated: true) {
            player.play()
        }
}
```

4. 비디오 재생 함수 만들기
앞의 두 액션 함수를 보면 url 을 만드는 내용만 다르고 그외의 내용은 동일하다. 이렇게 동일한 내용은 함수로 만들어 놓고 사용하는 것이 편리할 때가 많다. 따라서 이 코드를 '비디오 재생 함수'로 만들어 보자
url 을 사용하여 비디오를 재생하는 코드는 url 을 인수로 받는 playVidio 라는 함수를 만든다.

```
private func playVideo(url: NSURL)   {

}
```

5. 그런 다음 두 액션 함수에서 동일한 내용을 복사해 붙여 넣는다.

```
1   private func playVideo(url: NSURL)   {
2               let playerController = AVPlayerViewController()
3
4               let player = AVPlayer(url: url as URL)
5               playerController.player = player
6
7               self.present(playerController, animated: true) {
8                     player.play()
9               }
10  }
```

6. 두 액션 함수에서 동일하게 입력된 부분을 다음 코드로 대체한다. 이 코드는 url 을 얻은 후 playVidio 함수를 호출한다. 이렇게 하면 전체 소스가 훨씬 간략해지고 수정하기도 편리하다.

```
@IBAction func btnPlayInternalMovie(_ sender: UIButton) {
        // 내부 파일 mp4
        let filePath:String? = Bundle.main.path(forResource: "FastTyping", ofType:
"mp4")
        let url = NSURL(fileURLWithPath: filePath!)

        playVideo(url: url)
}


@IBAction func btnPlayExternalMovie(_ sender: UIButton) {
        // 외부 파일 mp4
        let url = NSURL(string:
"https://dl.dropboxusercontent.com/s/e38auz050w2mvud/Fireworks.mp4")!

        playVideo(url: url)
}
```