

# 멀티 뷰 기반의 프로그램

20167207 ZHENG JUN

# 01

## 탭 바 컨트롤러

### 1. 탭 바 컨트롤러 추가하기

Editor-> Embed in -> Tab Bar Controller

### 2. 탭 바 아이콘 변경하기

Attribute inspector ->System Item -> Top Related(Bookmarks,Custom)

### 탭 바에 뷰 추가하기

### 3. [뷰 컨트롤러 복사]

첫 번째 뷰 컨트롤러를 스토리보드에 복사하기

### 4. [파일 복사]

첫 번째 ViewController.swift를 네비게이션 영역에 복사한 후, 파일명 Ex1.swift과 클래스 이름 Ex1으로 변경하기

### 5. [뷰 컨트롤러와 파일 연결하기]

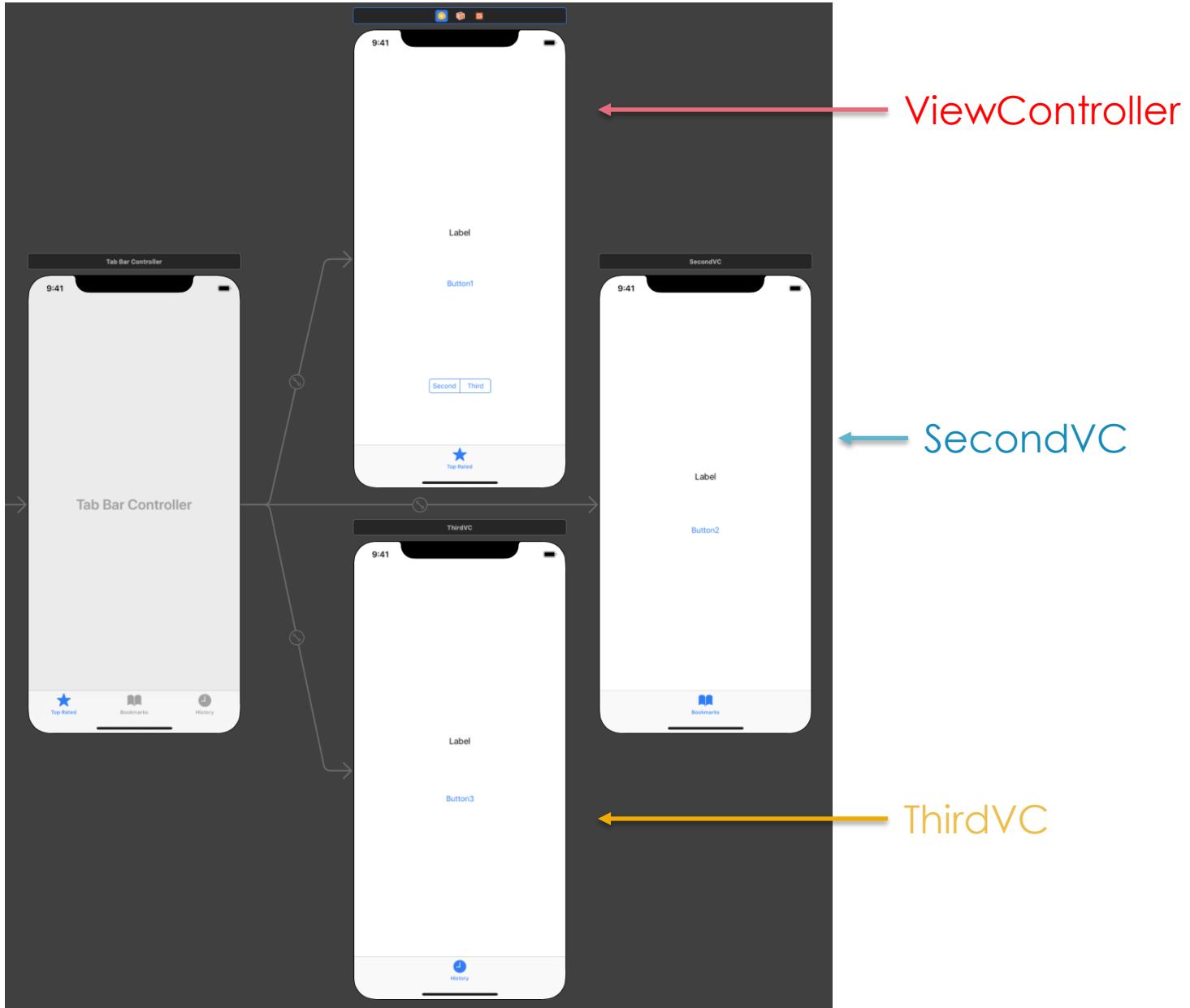
첫 번째 뷰 컨트롤러의 우측 상단(1)과 Identity inspector(2)를 클릭한 후, Class에서 Ex1을 선택(3)한다.

### 6. [뷰 컨트롤러와 탭 바 컨트롤러 연결]

탭 바 컨트롤러화면에 마우스 오른쪽 버튼을 클릭한 후, 첫 번째 뷰 컨트롤러화면으로 드래그한다.

01

# 탭 바 컨트롤러 ( 과제 1 )



```

9 import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBOutlet var lbl1: UILabel!
14     override func viewDidLoad() {
15         super.viewDidLoad()
16         // Do any additional setup after loading the view.
17     }
18
19     @IBAction func btn1(_ sender: UIButton) {
20         lbl1.text = "화면 1: 첫번째화면입니다."
21     }
22
23     // 세그먼트를 이용한 뷰 전환
24     @IBAction func sgChenge(_ sender: UISegmentedControl) {
25         if sender.selectedSegmentIndex == 0 { // 첫번째 세그먼트
26             tabBarController?.selectedIndex = 1 // 두번째 페이지
27         }else if sender.selectedSegmentIndex == 1 { // 두번째 세그먼트
28             tabBarController?.selectedIndex = 2 // 세번째 페이지
29         }
30     }
31 }
32 }
```

ViewController

```

9 import UIKit
10
11 class SecondVC: UIViewController {
12     @IBOutlet var lbl2: UILabel!
13     override func viewDidLoad() {
14         super.viewDidLoad()
15     }
16
17     @IBAction func btn2(_ sender: UIButton) {
18         lbl2.text = "화면 2: 첫번째화면입니다."
19     }
20 }
```

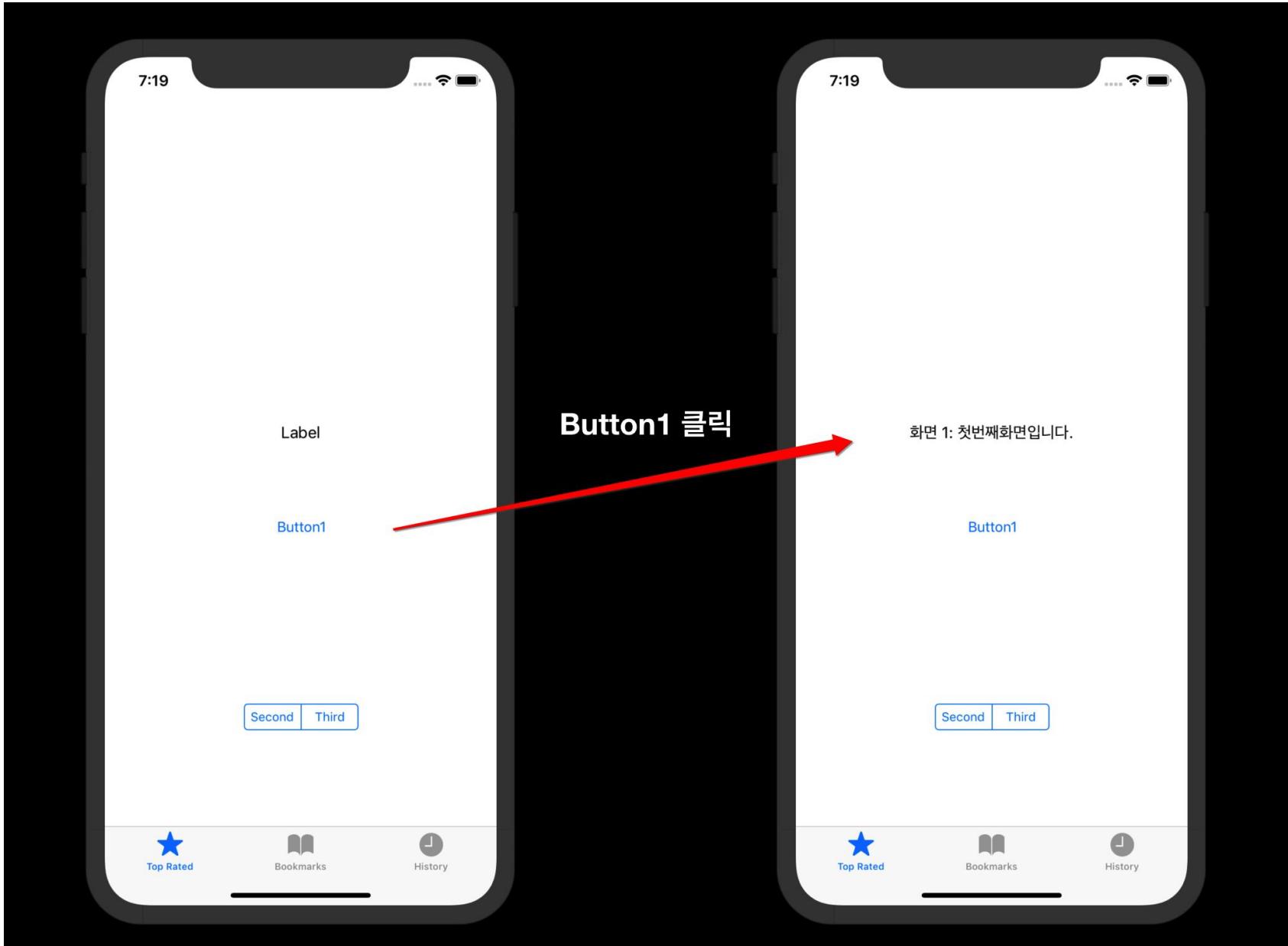
SecondVC

```

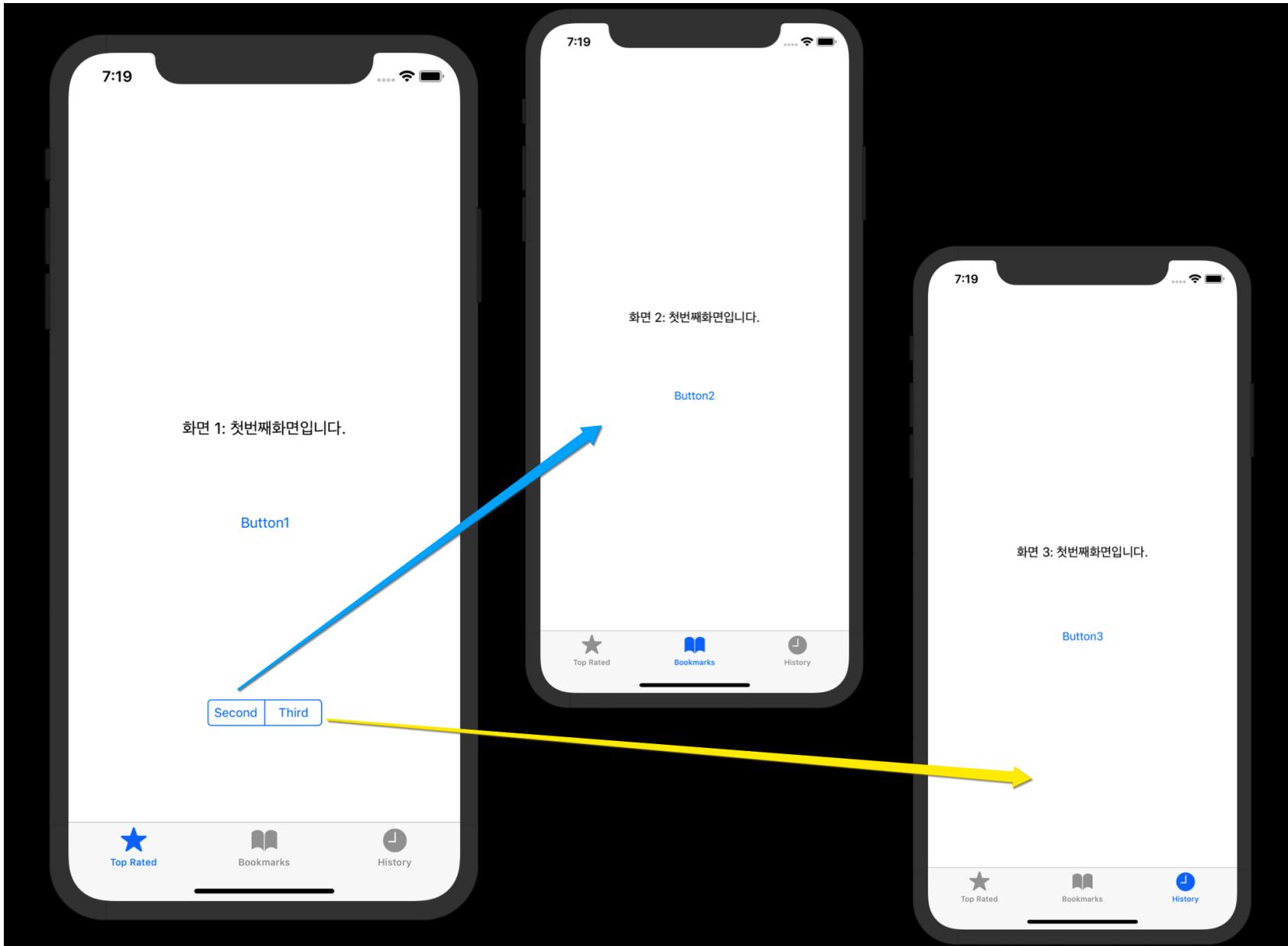
9 import UIKit
10
11 class ThirdVC: UIViewController {
12     @IBOutlet var lbl3: UILabel!
13     override func viewDidLoad() {
14         super.viewDidLoad()
15     }
16
17     @IBAction func btn3(_ sender: UIButton) {
18         lbl3.text = "화면 3: 첫번째화면입니다."
19     }
20 }
```

ThirdVC

# 01 텁 바 컨트롤러 ( 과제1 ) → 실행결과

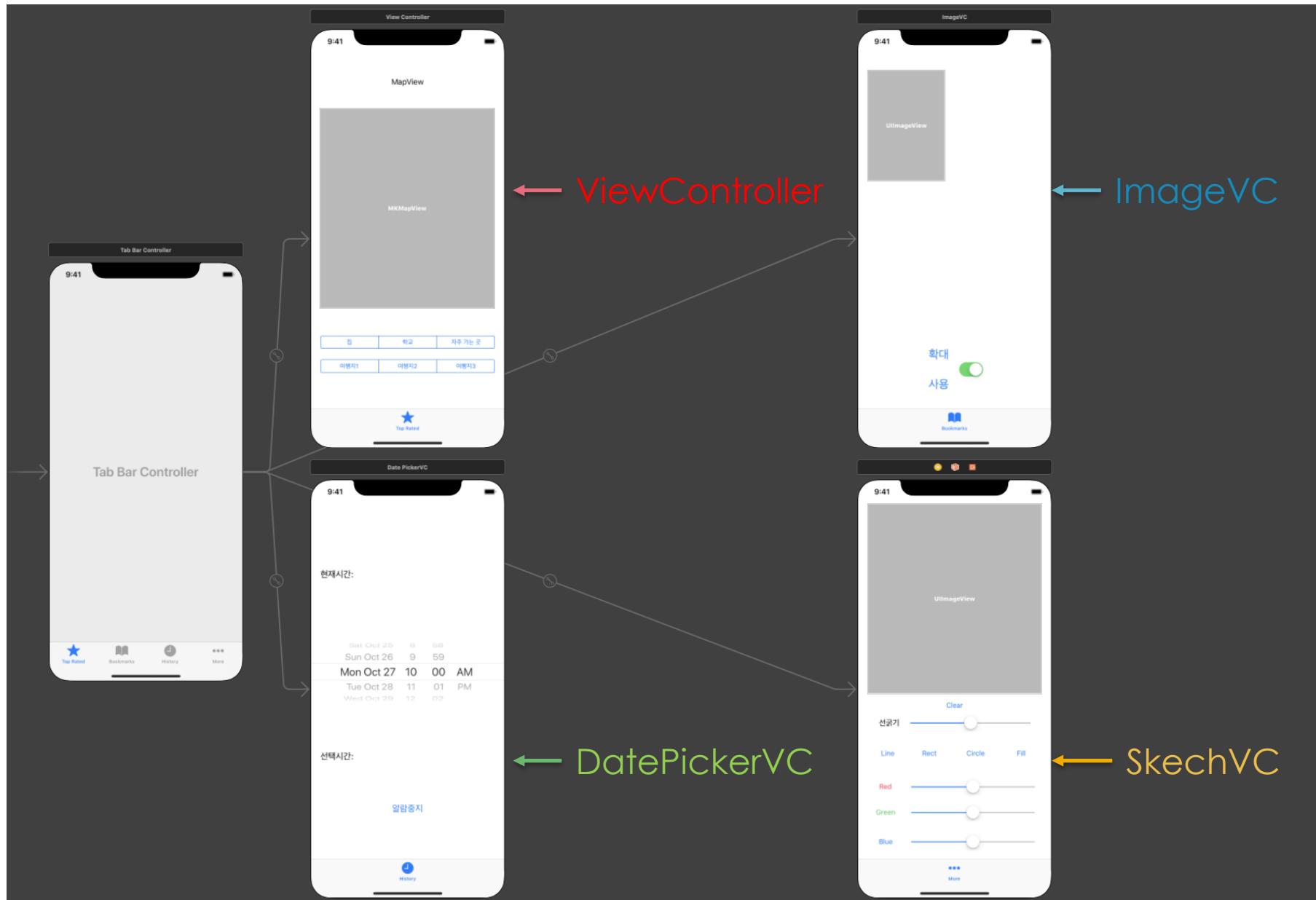


# 01 텁 바 컨트롤러 ( 과제1 ) → 실행결과



01

# 탭 바 컨트롤러 ( 과제2 )



```

8 //UIKit 이외에 지도를 사용하기 위해서 MapKit를 import 해야 한다
9
10 import UIKit
11 import MapKit
12
13 class ViewController: UIViewController {
14     // 아웃렛 변수
15     @IBOutlet var mapView: MKMapView!
16
17     override func viewDidLoad() {
18         super.viewDidLoad()
19     }

```

ViewController

```

9 import UIKit
10
11 class ImageVC: UIViewController {
12     var canUse: Bool?
13     //var isZoom = false
14     var imgOn: UIImage? //켜진 전구 이미지가 있는 UIImage 타입의 변수
15     var imgOff: UIImage? //꺼진 전구 이미지가 있는 UIImage 타입의 변수
16     var imgRemove: UIImage? //remove 이미지가 있는 UIImage 타입의 변수

```

ImageVC

```

8
9 import UIKit
10
11 class DatePickerVC: UIViewController {
12     @IBOutlet var btnAlarmStop: UIButton! //알람중지버튼 아웃렛 변수 추가
13     @IBOutlet var datePicker: UIDatePicker! //Picker 아웃렛 변수 추가
14
15     let timeSelector: Selector = #selector(DatePickerVC.updateTime)

```

DatePickerVC

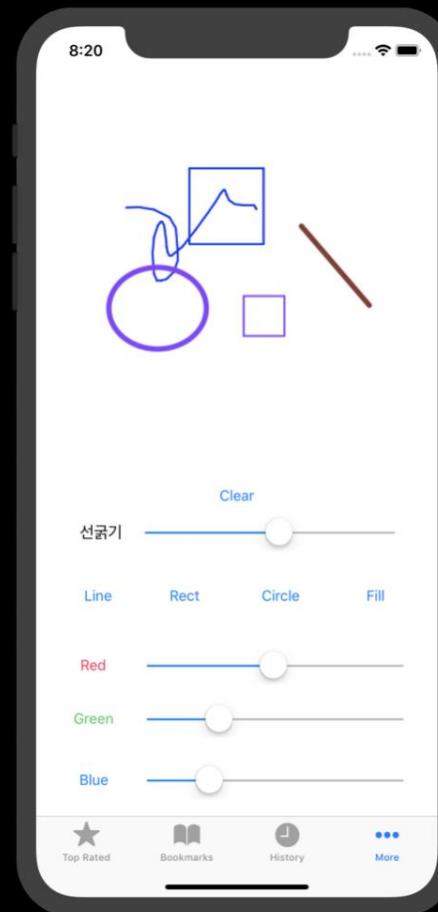
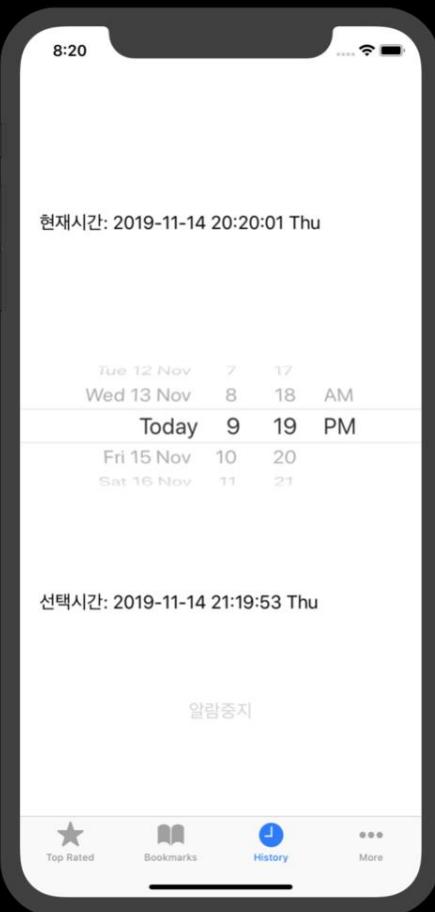
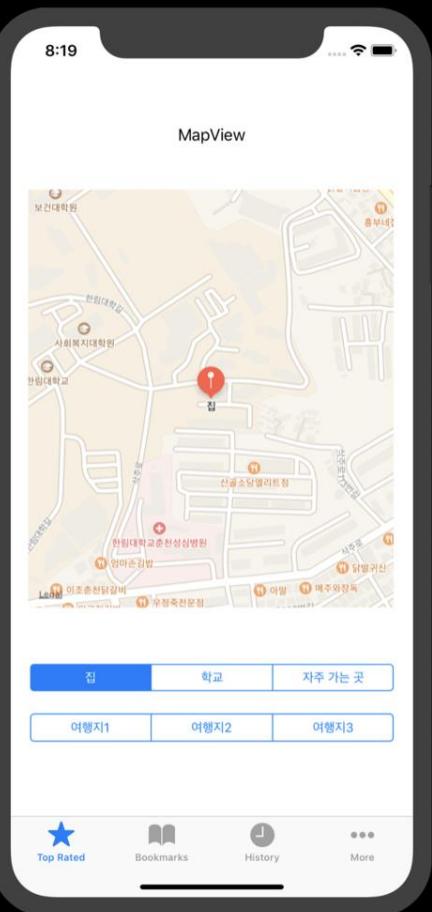
```

9 import UIKit
10
11 class SkechVC: UIViewController {
12     @IBOutlet var imgView: UIImageView!
13     @IBOutlet var sliderW: UISlider!
14     @IBOutlet var sliderRed: UISlider!
15     @IBOutlet var sliderGreen: UISlider!
16     @IBOutlet var sliderBlue: UISlider!
17
18     var lastPoint: CGPoint!
19     var firstPoint: CGPoint!

```

SkechVC

# 01 텁 바 컨트롤러 ( 과제2 ) → 실행결과



Main화면 ( ViewController )

이미지 뷰 ( ImageVC )

데이터 피커 ( DatePickerVC )

스케치 ( SketchVC )

# 02

내비게이션 컨트롤러

# 내비게이션 컨트롤러

- 내비게이션 컨트롤러 추가하기
- 뷰 컨트롤러의 제목 변경하기

탭 바에 뷰 추가하기

- [새로운 뷰 컨트롤러 추가]
- 새로운 뷰 컨트롤러의 제목 변경하기
- [바 버튼으로 화면1과 화면2 연결하기]

(1) [화면1에서 바 버튼 만들기]

(2) [바 버튼과 화면 2 연결하기]

화면 1의 바 버튼을 마우스 오른 쪽 버튼을 누른 상태에서 화면 2로 드래그하면, **Action Segue**의 검은 창이 나타나는데 **Show**를 선택한다.

(3) [세그웨이 이름 변경하기]

두 화면을 연결한 세그웨이 그림(1)을 선택한 후, Attribute inspector -> Identifier -> **sg1**

(4) [바 버튼의 이름 변경하기]

바 버튼을 누른 후, Attribute inspector -> System Item -> **Edit**(Add, Done, Cancel, Save)

- [버튼으로 화면1과 화면2 연결하기]

(1) [화면1에서 바 버튼 만들기]

(2) [버튼과 화면 2 연결하기]

(3) [세그웨이 이름 변경하기]

- [화면 2의 소스 파일 만들기]

(1) 메뉴에서 File -> New -> File... 을 선택한 후,

나타나는 창에서 iOS -> Source -> **Cocoa Touch Class**를 선택하고 Next를 클릭한다.

(2) 창이 나타나면, Class에 **VC2**로 하고, Subclass of는 **UIViewController**로 입력하고 Next를 클릭한다.

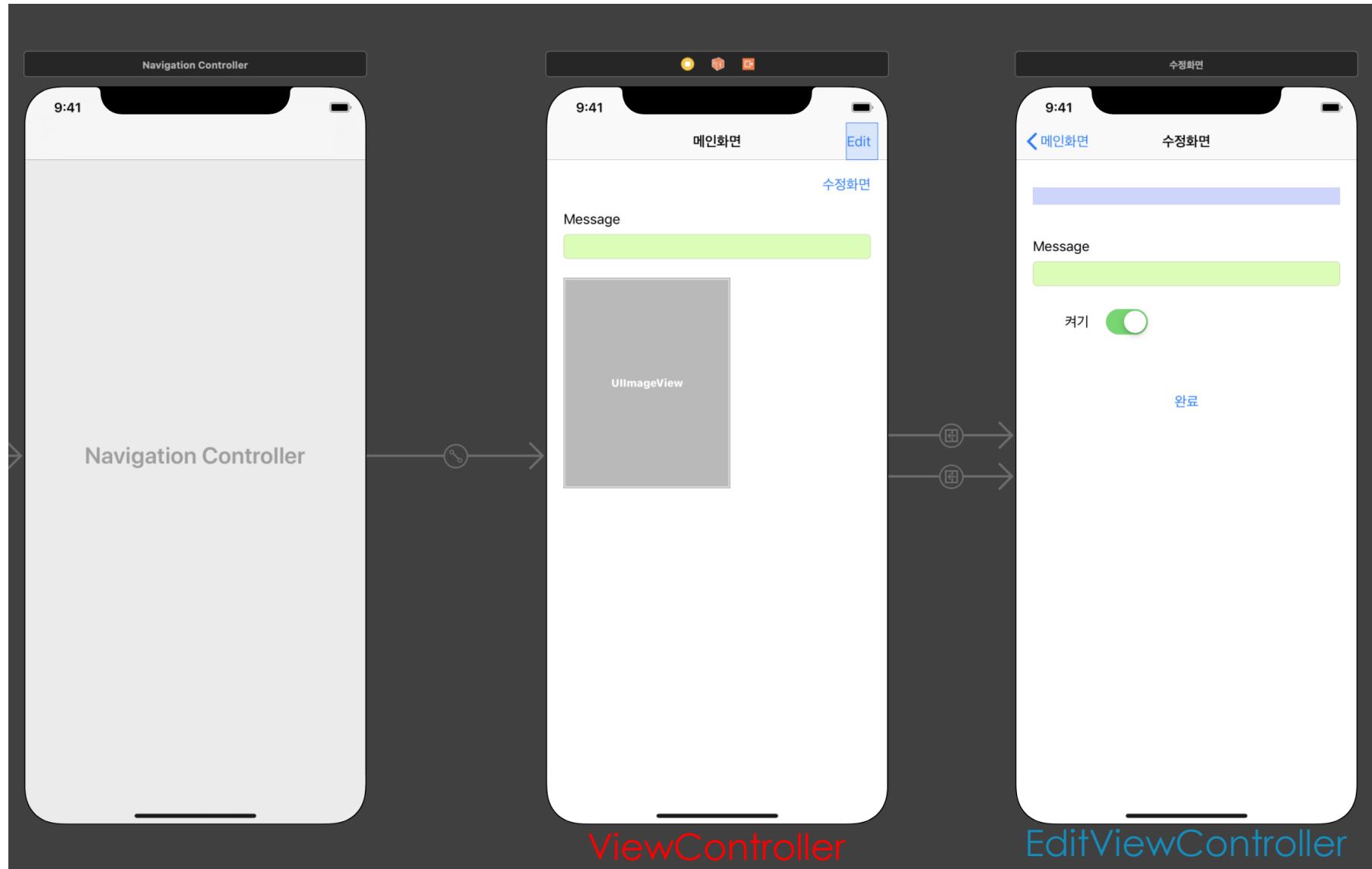
(3) ViewController2.swift에서 클래스 이름 **VC2**으로 변경한다.

- [화면2와 소스파일 연결하기]

화면2의 우측 상단(1)과 Identity inspector(2)를 클릭한 후, Class에서 **VC2**를 선택(3)한다.

02

## 내비게이션 컨트롤러 ( 과제3 )



## ViewController

```
class ViewController: UIViewController, EditDelegate { // 프로토콜 EditDelegate를 상속받는다

    let imgOn = UIImage(named: "lamp-on.png")
    let imgOff = UIImage(named: "lamp-off.png")
    var isOn1 = true
```

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    // 수정화면 객체 생성
    let editViewController = segue.destination as! EditViewController
    // if문과 else문을 이용해 세그웨이 별로 다른 문자열을 전송한다.
    if segue.identifier == "editButton" { // 버튼을 클릭한 경우
        editViewController.lblTextWay = "EditButton2"
    } else if segue.identifier == "editBarButton" { // 바 버튼을 클릭한 경우
        editViewController.lblTextWay = "EditBarButton1"
    }
    editViewController.textMessage = txMessage.text!
    editViewController.isOn2 = isOn1
    editViewController.delegate = self // 위임자를 설정
}

// protocol을 이용하여 데이터를 송신측에서 수신측으로 전달하기 위해
// 메인 프로그램(수신측)의 함수를 구현한다
func didMessageEditDone(_ controller: EditViewController, message: String) {
    txMessage.text = message
}

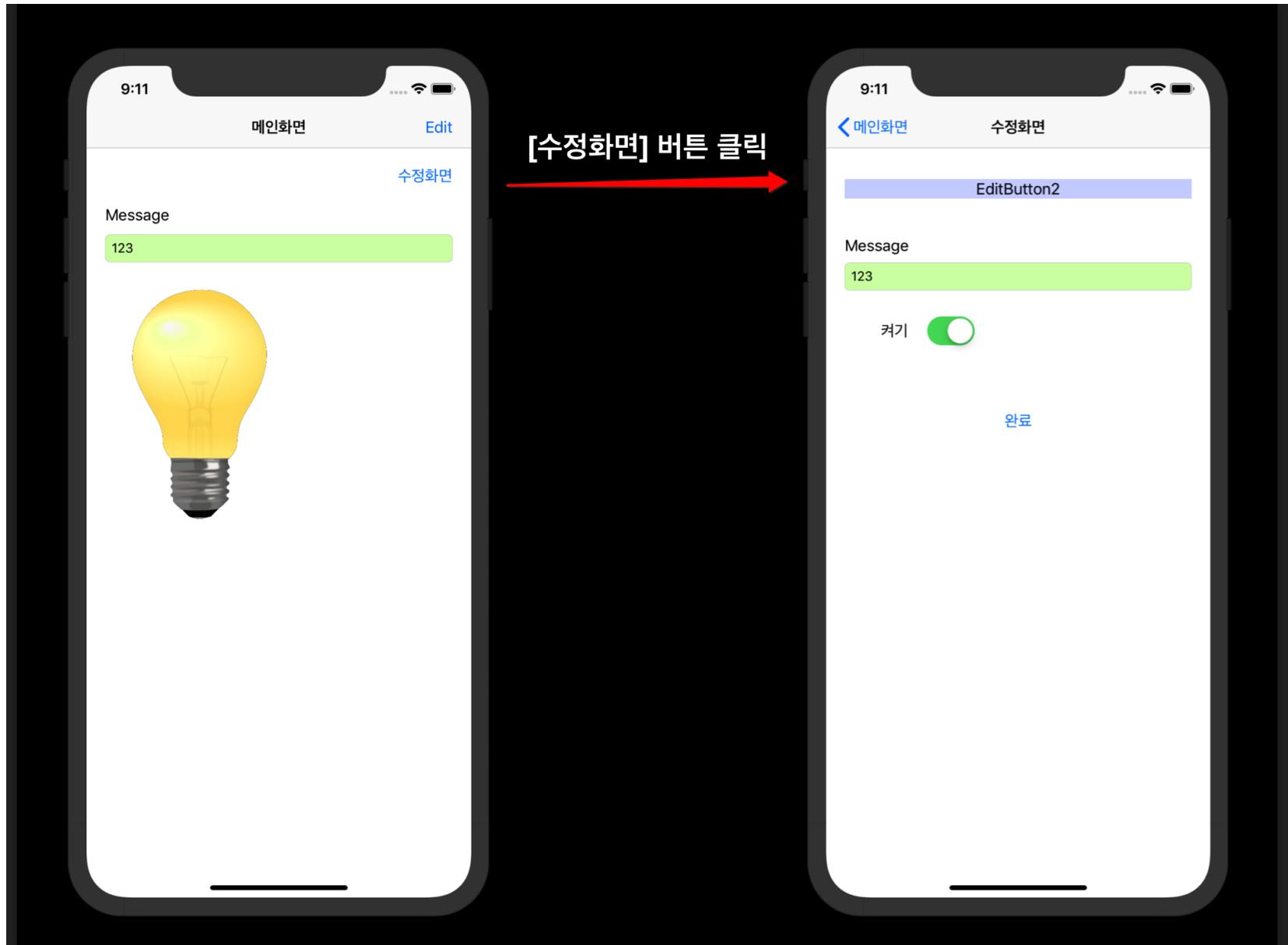
func didImageOnOffDone(_ controller: EditViewController, isOn: Bool) {
    if isOn {
        imgView.image = imgOn
        isOn1 = true
    } else {
        imgView.image = imgOff
        isOn1 = false
    }
}
```

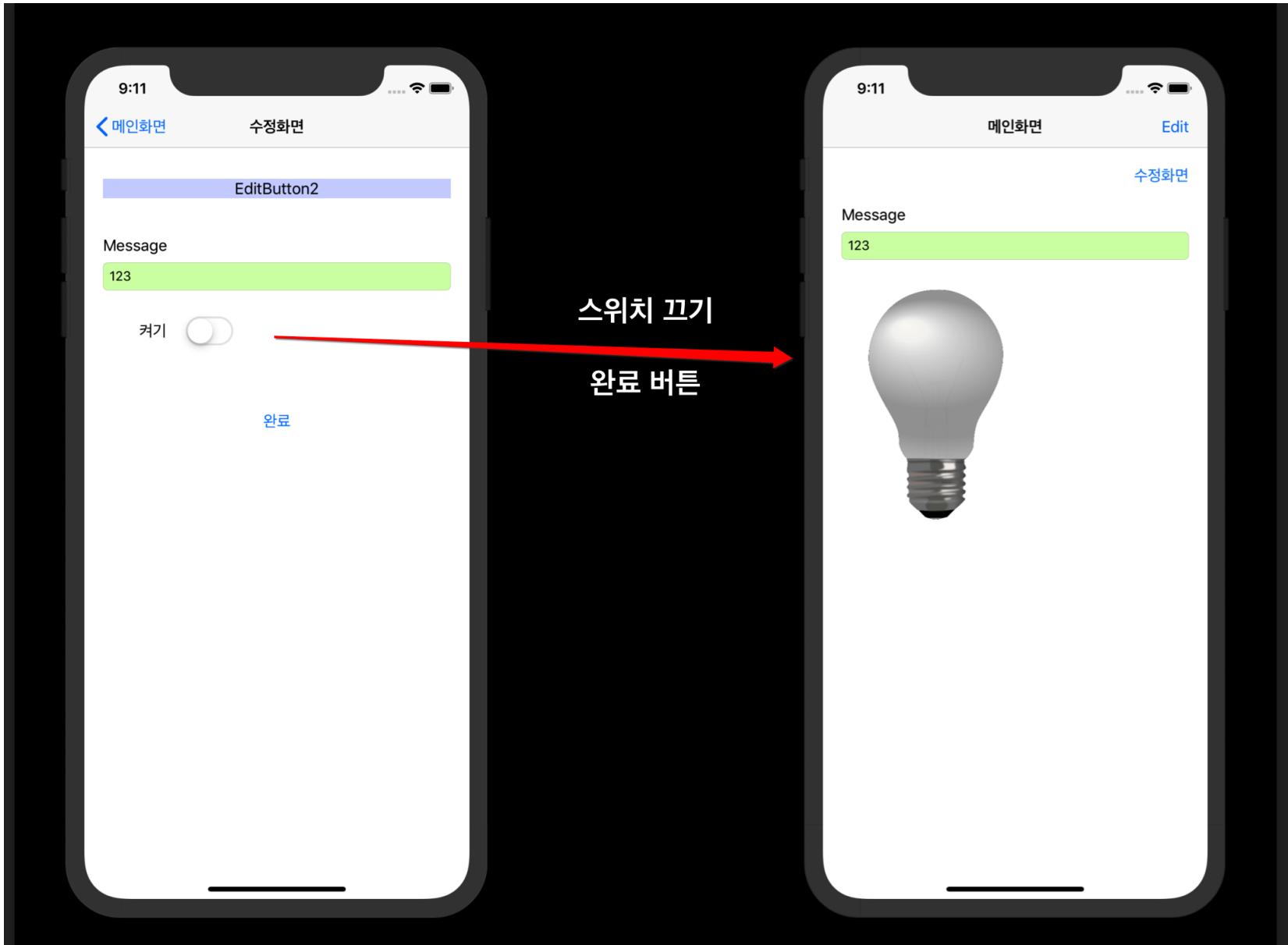
## EditViewController

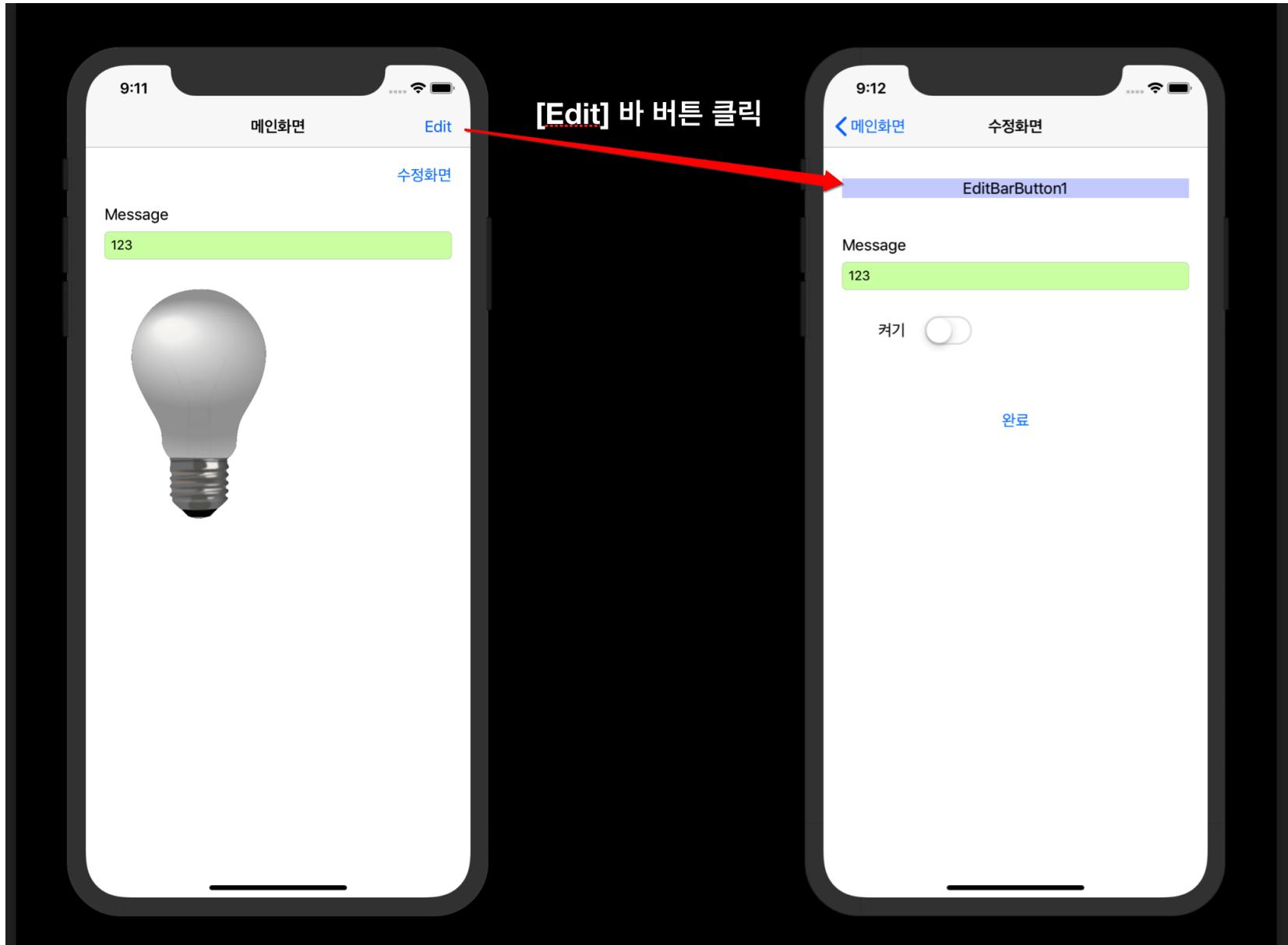
```
11 // 수정화면(송신측) 프로토콜은 메인 화면의 함수를 구현하지 않고 선언만 한다.  
12 protocol EditDelegate {  
13     func didMessageEditDone(_ controller: EditViewController, message: String)  
14     func didImageOnOffDone(_ controller: EditViewController, isOn: Bool)  
15 }
```

```
19     var lblTextWay: String = ""  
20     var textMessage: String = ""  
21     var delegate : EditDelegate? // delegate 변수를 생성한다  
22     var isOn2 = false  
23     @IBOutlet var txMessage: UITextField!  
24     @IBOutlet var lblWay: UILabel!  
25     @IBOutlet var swIsOn: UISwitch!  
26     override func viewDidLoad() {  
27         super.viewDidLoad()  
28         // 초기화  
29         // 메인화면에서 받은 값을 출력한다  
30         lblWay.text = lblTextWay  
31         txMessage.text = textMessage  
32         swIsOn.isOn = isOn2  
33     }
```

```
@IBAction func btnDone(_ sender: UIButton) {  
    if delegate != nil {  
        // 함수 호출  
        delegate?.didMessageEditDone(self, message: txMessage.text!)  
        delegate?.didImageOnOffDone(self, isOn: isOn2)  
    }  
    // 세그웨이를 사용하지 않고 메인화면으로 되돌아가는 방법  
    _ = navigationController?.popViewController(animated: true)  
}
```

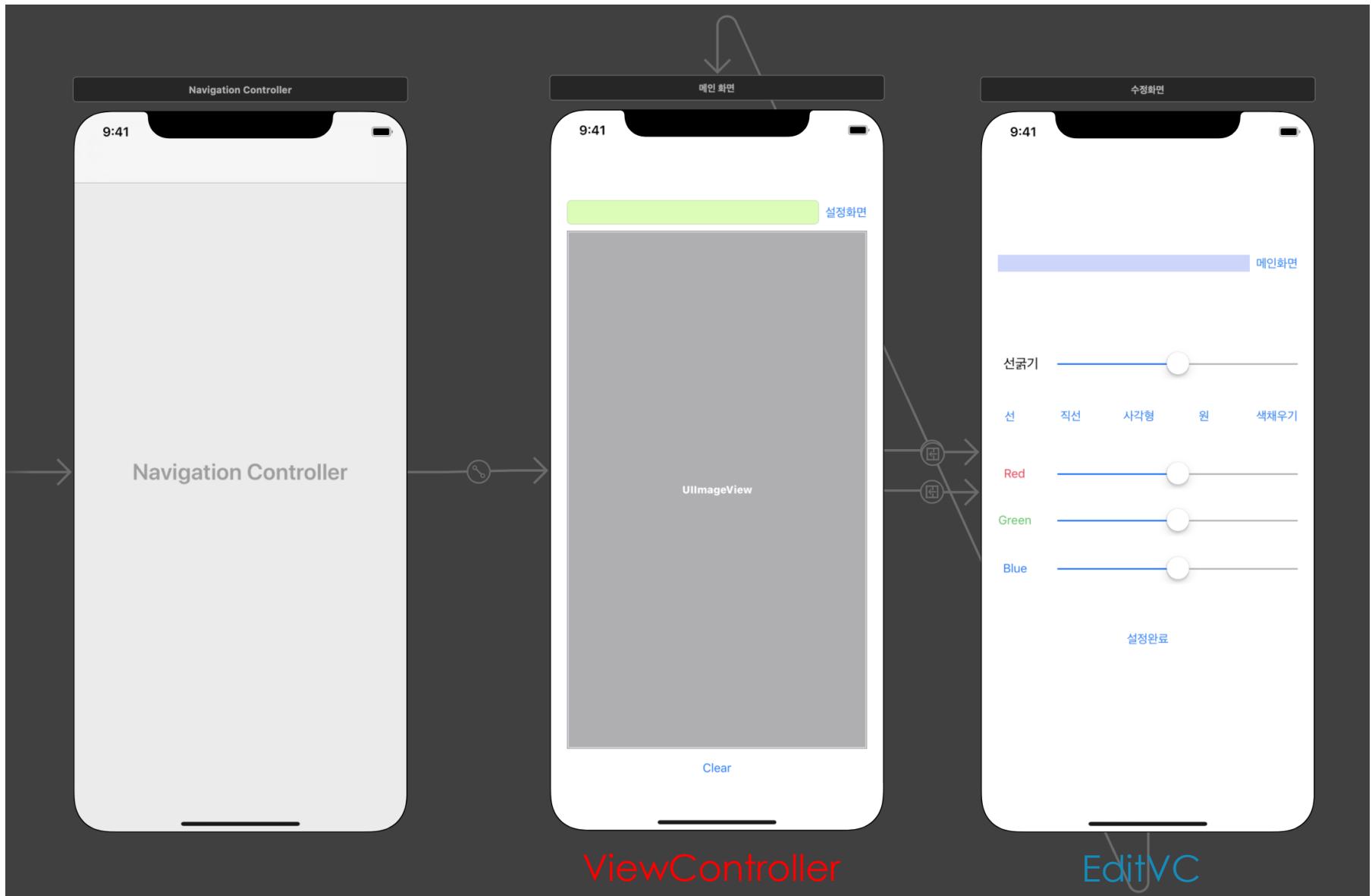






02

## 내비게이션 컨트롤러 ( 과제4 )



## ViewController

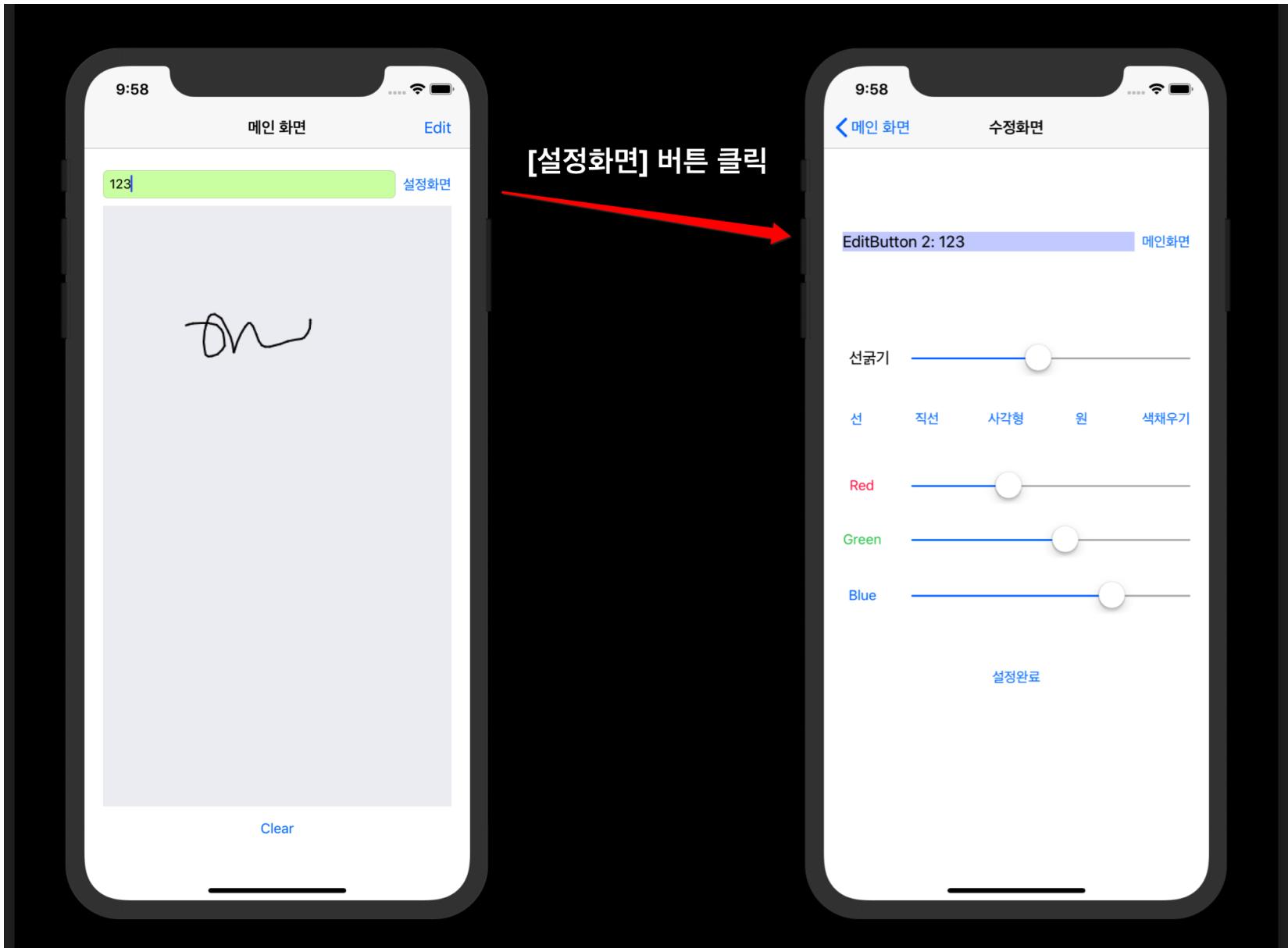
```
var fillType: Bool = false // 채우기 여부 true -> 채우기

override func viewDidLoad() {
    super.viewDidLoad()
    let myColor = UIColor(displayP3Red:
        CGFloat(sliderRed1), green: CGFloat(sliderGreen1), blue:
        CGFloat(sliderBlue1), alpha: 1.0)
    lineColor = myColor.cgColor
}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    let str = txtMain.text!
    // 수정화면 객체 생성
    let editViewController = segue.destination as! EditVC
    // if문과 else문을 이용해 세그웨이 별로 다른 문자열을 전송한다.
    if segue.identifier == "editButton" { // 버튼을 클릭한 경우
        editViewController.bufferEdit = "EditButton 2: " + str
    } else if segue.identifier == "editBarButton" { // 바 버튼을 클릭한 경우
        editViewController.bufferEdit = "EditBarButton 1: " + str
    }
    // 메인화면 -> 수정화면
    editViewController.Red2 = sliderRed1
    editViewController.Green2 = sliderGreen1
    editViewController.Blue2 = sliderBlue1
    editViewController.Width2 = lineSize
    editViewController.sketchType2 = sketchType
    editViewController.fillType2 = fillType
}
```

## EditVC

```
var fillType2: Bool = false
override func viewDidLoad() {
    super.viewDidLoad()
    sliderWidth.maximumValue = 10.0
    sliderWidth.minimumValue = 0.0
    sliderWidth.value = Float(Width2)
    sliderRed.maximumValue = 1.0
    sliderRed.minimumValue = 0.0
    sliderRed.value = Float(Red2)
    sliderGreen.maximumValue = 1.0
    sliderGreen.minimumValue = 0.0
    sliderGreen.value = Float(Green2)
    sliderBlue.maximumValue = 1.0
    sliderBlue.minimumValue = 0.0
    sliderBlue.value = Float(Blue2)
    lblEdit.text = bufferEdit
}
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    // 메인화면 객체 생성
    let viewController = segue.destination as! ViewController
    // 메인화면 버튼을 누르면 데이터 전송 수정화면 -> 메인화면
    if segue.identifier == "mainButton" {
        viewController.sliderRed1 = CGFloat(sliderRed.value)
        viewController.sliderGreen1 = CGFloat(sliderGreen.value)
        viewController.sliderBlue1 = CGFloat(sliderBlue.value)
        viewController.lineSize = CGFloat(sliderWidth.value)
        viewController.sketchType = sketchType2
        viewController.fillType = fillType2
    }
}
@IBAction func btnFillColor(_ sender: UIButton) {
    fillType2 = !fillType2
}
```





02

## 내비게이션 컨트롤러 ( 과제5 )



## ViewController

```
override func viewDidLoad() {
    super.viewDidLoad()
    let myColor = UIColor(displayP3Red:
        CGFloat(sliderRed1), green: CGFloat(sliderGreen1), blue: CGFloat(sliderBlue1), alpha: 1.0)
    lineColor = myColor.cgColor
}

// protocol을 이용하여 데이터를 수신측에서 수신측으로 전달하기 위해
// 메인 프로그램(수신측)의 함수를 구현한다
func fooSetLineColor(_ red2: CGFloat, green2: CGFloat, blue2: CGFloat){ // 받아온 컬러 값을 저장하는 함수
    let myColor = UIColor(displayP3Red:
        CGFloat(red2), green: CGFloat(green2), blue: CGFloat(blue2), alpha: 1.0)
    sliderRed1 = red2
    sliderBlue1 = blue2
    sliderGreen1 = green2
    lineColor = myColor.cgColor
}

func fooSetLineWidth(_ width2: CGFloat){ // 받아온 lineWidth를 저장하는 함수
    lineSize = width2
}

func fooSet(_ sketchType2: Int, fillType2: Bool){ // 받아온 sketchType, fillType를 저장하는 함수
    sketchType = sketchType2
    fillType = fillType2
}
```

## ViewController

```
import UIKit
class ViewController: UIViewController, EditDelegate {
    @IBOutlet var txtMain: UITextField!
    @IBOutlet var imgView: UIImageView!
    var lastPoint: CGPoint!
```

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    let str = txtMain.text!
    // 수정화면 객체 생성
    let editViewController = segue.destination as! EditVC
    // if문과 else문을 이용해 세그웨이 별로 다른 문자열을 전송한다.
    if segue.identifier == "editButton" { // 버튼을 클릭한 경우
        editViewController.bufferEdit = "EditButton 2: " + str
    } else if segue.identifier == "editBarButton" { // 바 버튼을 클릭한 경우
        editViewController.bufferEdit = "EditBarButton 1: " + str
    }
    // 메인화면 -> 수정화면
    editViewController.delegate = self // protocol 사용하기 위해 위임자 설정
    editViewController.Red2 = sliderRed1
    editViewController.Green2 = sliderGreen1
    editViewController.Blue2 = sliderBlue1
    editViewController.Width2 = lineSize
    editViewController.sketchType2 = sketchType
    editViewController.fillType2 = fillType
}
```

## EditVC

```
protocol EditDelegate {
    func fooSetColor(_ red2: CGFloat, green2: CGFloat, blue2: CGFloat)
    func fooSetLineWidth(_ width2: CGFloat)
    func fooSet(_ sketchType2: Int, fillType2: Bool)
}
```

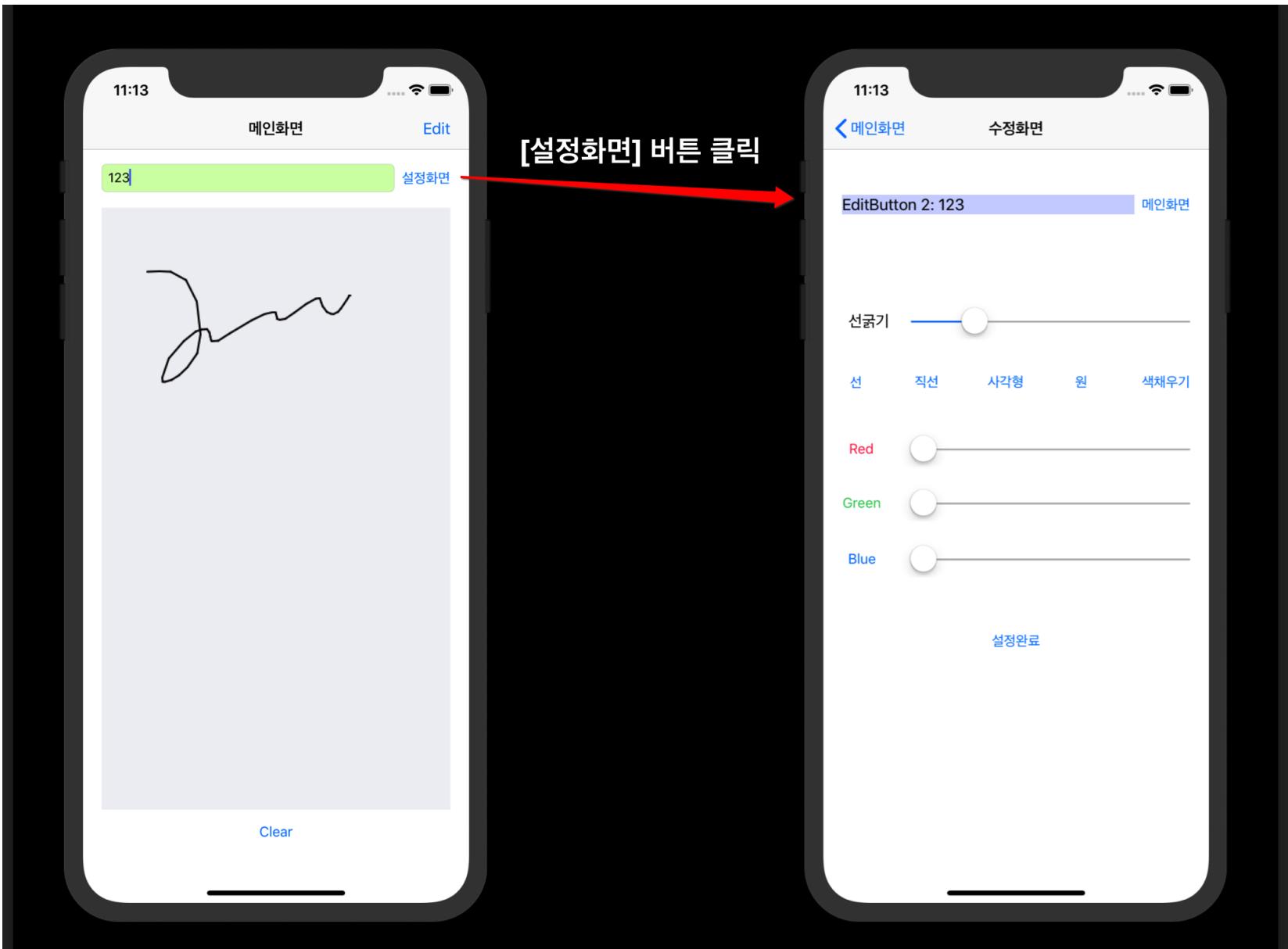
```
class EditVC: UIViewController {
```

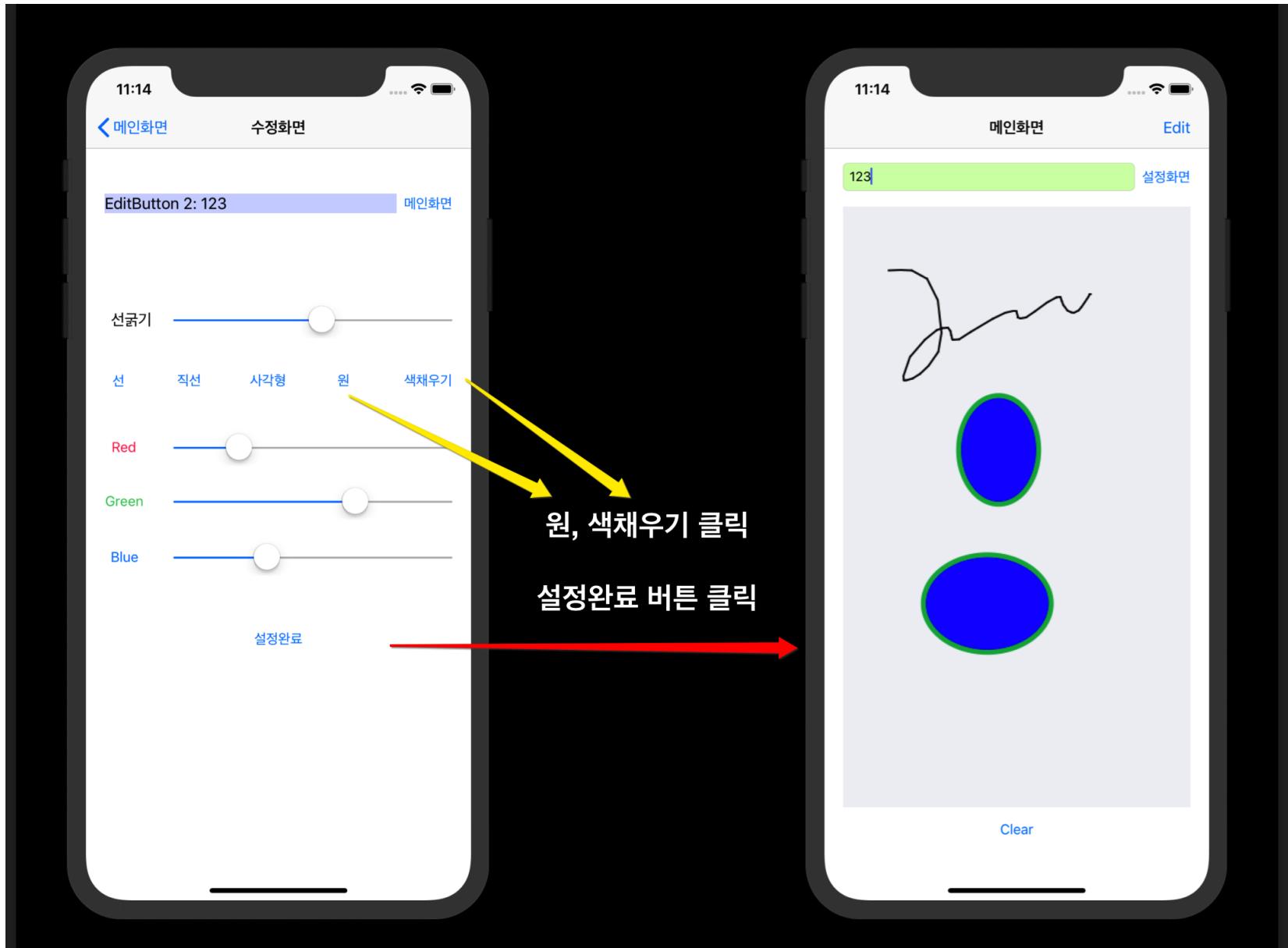
```
var delegate: EditDelegate? // delegate 변수를 생성한다
```

```
@IBAction func btnDone(_ sender: UIButton) {
    Red2 = CGFloat(sliderRed.value)
    Green2 = CGFloat(sliderGreen.value)
    Blue2 = CGFloat(sliderBlue.value)
    Width2 = CGFloat(sliderWidth.value)
    // 함수 호출
    if delegate != nil{
        delegate?.fooSetColor(Red2, green2: Green2, blue2: Blue2)
        delegate?.fooSetLineWidth(Width2)
        delegate?.fooSet(sketchType2, fillType2: fillType2)
    }
    // 세그웨이를 사용하지 않고 메인화면으로 되돌아가는 방법
    _ = navigationController?.popViewController(animated: true)
}
```

## EditVC

```
var fillType2: Bool = false
override func viewDidLoad() {
    super.viewDidLoad()
    sliderWidth.maximumValue = 10.0
    sliderWidth.minimumValue = 0.0
    sliderWidth.value = Float(Width2)
    sliderRed.maximumValue = 1.0
    sliderRed.minimumValue = 0.0
    sliderRed.value = Float(Red2)
    sliderGreen.maximumValue = 1.0
    sliderGreen.minimumValue = 0.0
    sliderGreen.value = Float(Green2)
    sliderBlue.maximumValue = 1.0
    sliderBlue.minimumValue = 0.0
    sliderBlue.value = Float(Blue2)
    lblEdit.text = bufferEdit
}
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    // 메인화면 객체 생성
    let viewController = segue.destination as! ViewController
    // 메인화면 버튼을 누르면 데이터 전송 수정화면 -> 메인화면
    if segue.identifier == "mainButton" {
        viewController.sliderRed1 = CGFloat(sliderRed.value)
        viewController.sliderGreen1 = CGFloat(sliderGreen.value)
        viewController.sliderBlue1 = CGFloat(sliderBlue.value)
        viewController.lineSize = CGFloat(sliderWidth.value)
        viewController.sketchType = sketchType2
        viewController.fillType = fillType2
    }
}
@IBAction func btnFillColor(_ sender: UIButton) {
    fillType2 = !fillType2
}
```





감사합니다  
**THANK YOU  
FOR WATCHING**