[IGRUS]
iOS 스터디

FINAL / 2020.08.25

12171571 강민상 / FDEE

오늘의 목표

Xcode 짤막한 지식

- 외부 글씨체 적용방법

12171571 강민상 / FDEE

오늘의 목표

Xcode Swift 프로그래밍

- 프로토콜 구현
- 백그라운드 구현 -> 1차 완성!!
- 빡공률 구현 - 도전!
- 프로그래스 구현 - 도전!

12171571 강민상 / FDEE

Xcode

Xcode 짤막한 지식

- 외부 글씨체 적용방법
- 1. .otf or .ttf 글씨체 파일을 준비한다
- 2. 글씨체 파일을 폴더에 넣어 프로젝트 폴더에 추가한다
- 3. Xcode 내에서 폴더를 생성하여 드래그로 글씨체 파일을 추가한다
- 4. Info.plist -> Fonts provided by application에 파일명을 추가한다
- 5. 잘 따라했다면 글씨체가 적용된다

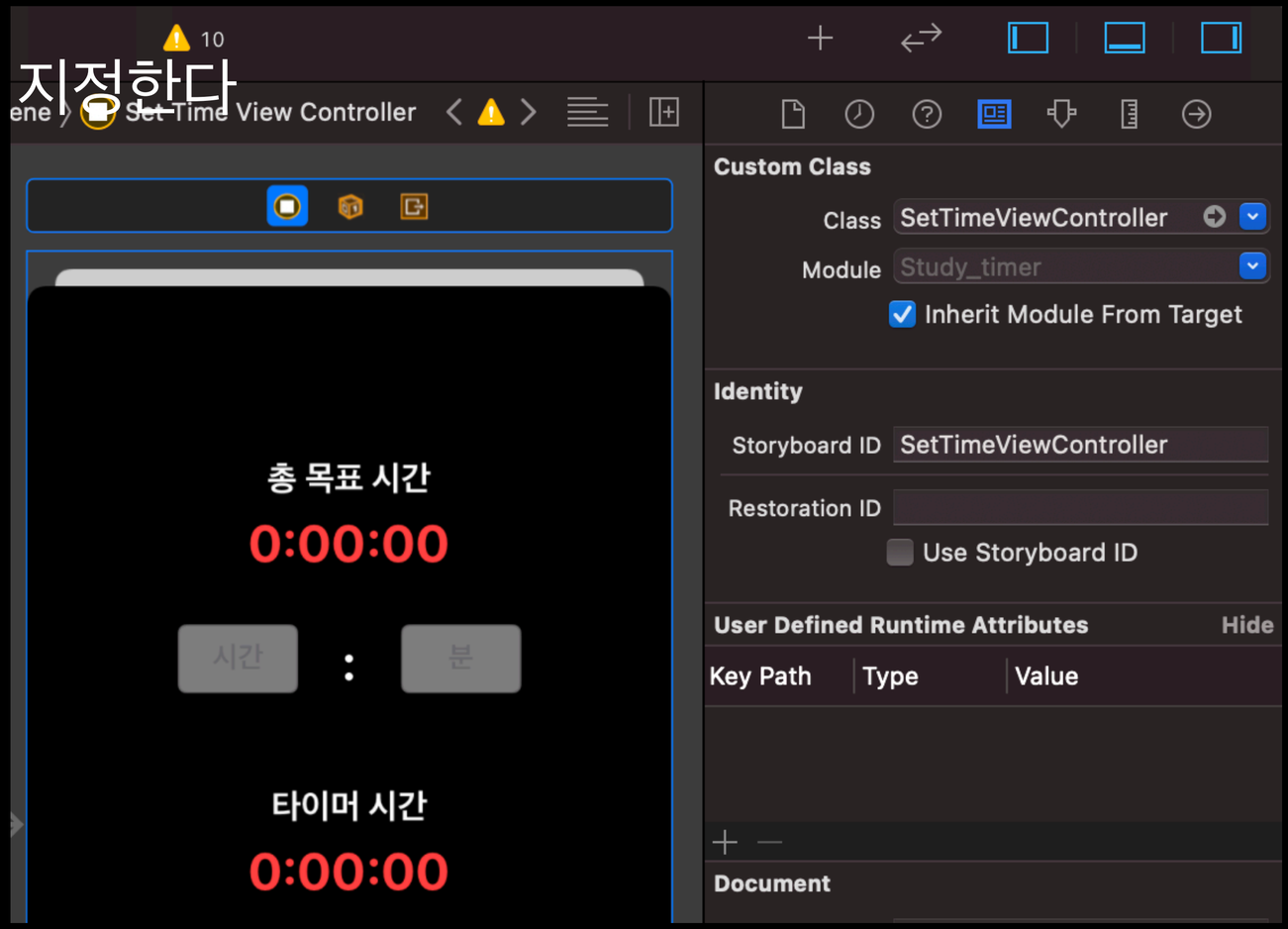
~~※ 단, 글씨체 적용이 안되는 경우도 발생한다 (이유는 모른다...)~~

관련글 : https://codewithchris.com/common-mistakes-with-adding-custom-fonts-to-your-ios-app/

Xcode

Xcode Swift 프로그래밍

- 프로토콜 구현 (저도 잘 모르는 부분이라 일단 구현하는 방향으로...)
- 1. StoryBoard ID 값을 지정한다



Xcode

Xcode Swift 프로그래밍

- 프로토콜 구현

- 2. ViewController에 코드를 추가한다

```
extension ViewController : ChangeViewController {

    func updateViewController() {
        getTimeData()
        updateShow()
        stopColor()
        stopEnable()
        Button_RESTART_Outlet.backgroundColor = BUTTON_CLICK
        Button_RESTART_Outlet.isUserInteractionEnabled = false
    }
}
```

```
229
230 }
231
232 extension ViewController : ChangeViewController {
233
234     func updateViewController() {
235         getTimeData()
236         updateShow()
237         stopColor()
238         stopEnable()
239         Button_RESTART_Outlet.backgroundColor = BUTTON_CLICK
240         Button_RESTART_Outlet.isUserInteractionEnabled = false
241     }
242 }
243
244
```

12171571 강민상 / FDEE

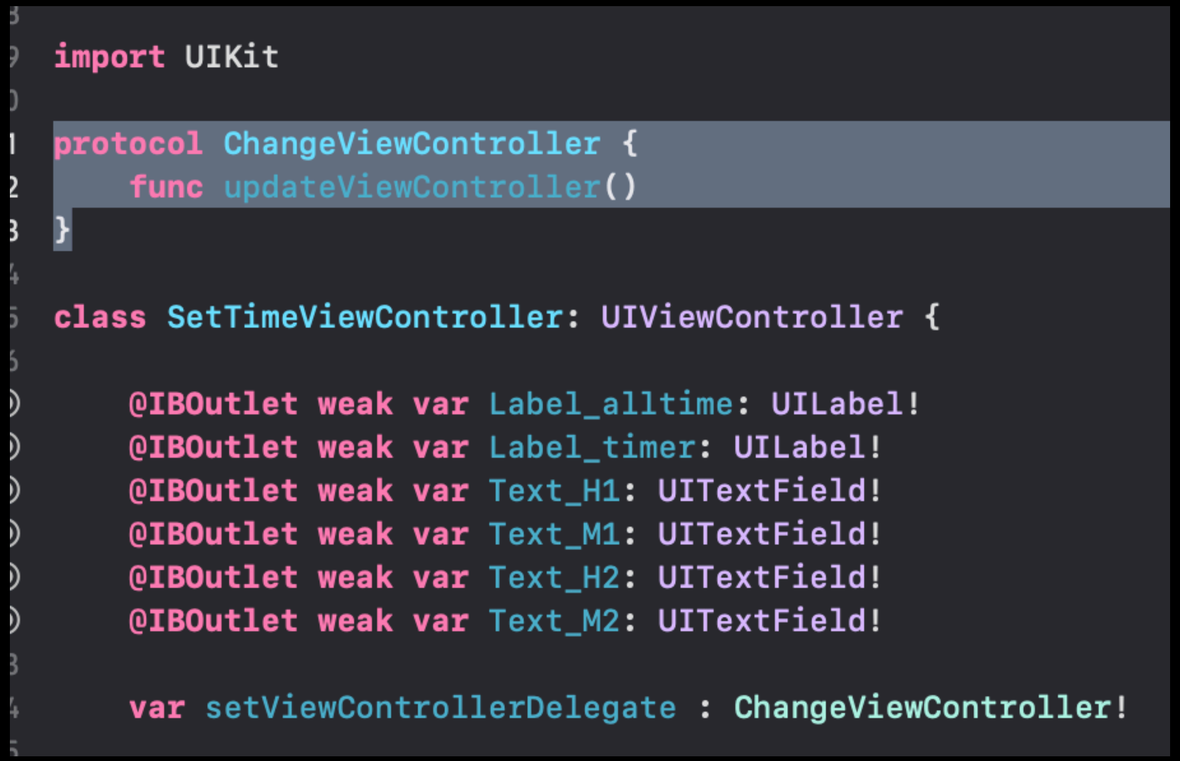
Xcode

Xcode Swift 프로그래밍

- 프로토콜 구현

- 3. SetTimeViewController에 protocol을 명시한다

```
protocol ChangeViewController {
    func updateViewController()
}
```



12171571 강민상 / FDEE

Xcode

Xcode Swift 프로그래밍

- 프로토콜 구현

- 4. SetTimeViewController에 변수를 만든다

```
var setViewControllerDelegate : ChangeViewController!
```

```
import UIKit

protocol ChangeViewController {
    func updateViewController()
}

class SetTimeViewController: UIViewController {

    @IBOutlet weak var Label_alltime: UILabel!
    @IBOutlet weak var Label_timer: UILabel!
    @IBOutlet weak var Text_H1: UITextField!
    @IBOutlet weak var Text_M1: UITextField!
    @IBOutlet weak var Text_H2: UITextField!
    @IBOutlet weak var Text_M2: UITextField!

    var setViewControllerDelegate : ChangeViewController!
```

12171571 강민상 / FDEE

Xcode

Xcode Swift 프로그래밍

- 프로토콜 구현

- 5. SetTimeViewController에 Button_SET Action에 코드를 추가한다

`setViewControllerDelegate.updateViewController()`

```
@IBAction func Button_SET(_ sender: UIButton) {
    allTime = H1*3600 + M1*60
    second = H2*3600 + M2*60
    UserDefaults.standard.set(allTime, forKey: "allTime")
    UserDefaults.standard.set(second, forKey: "second")
    setViewControllerDelegate.updateViewController()
    self.dismiss(animated: true, completion: nil)
}
```

12171571 강민상 / FDEE

Xcode

Xcode Swift 프로그래밍

- 프로토콜 구현

- 6. ViewController에 Button_TimeSET Action에 코드를 추가한다

```
        let setVC = storyboard?.instantiateViewController(withIdentifier: "SetTimeViewController") as! SetTimeViewController
        setVC.setViewControllerDelegate = self
        present(setVC,animated: true,completion: nil)
```

```
    @IBAction func Button_TimeSET_Action(_ sender: UIButton) {
        let setVC = storyboard?.instantiateViewController(withIdentifier: "SetTimeViewController") as!
            SetTimeViewController
        setVC.setViewControllerDelegate = self
        present(setVC,animated: true,completion: nil)
    }
```

12171571 강민상 / FDEE

Xcode

Xcode Swift 프로그래밍

- 프로토콜 구현 대략적인 설명 (저도 어려워요ㅠㅠ)
- ViewControlller의 extension을 활용하여 ChangeViewControlller 라는 ViewControlller를 만든다
- ChangeViewControlller에 필요한 메소드인 updateViewControlller()를 만든다
- 셋팅화면 (다른화면)에 protocol를 명시한다
- 셋팅화면에서 사용하기위해 ChangeViewControlller 타입의 변수 SetViewControllerDelegate 를 만든다
- 사용은 변수.메소드 형식으로 updateVoewControlller()가 실행되는 식이다
- 마지막으로 TimeSET 버튼 클릭시에 ViewControlller의 setVC와 세팅화면의 변수 SetViewControllerDelegate 를 동일시 시켜 연결고리?를 만들어주는 느낌이다

Xcode

Xcode Swift 프로그래밍

- 백그라운드 구현 - 알고리즘
- 화면을 나갔을시 : 화면을 나간 시간을 임시로 따로 저장한다
- 되돌아왔을시 : 들어온 시간을 따로 저장한다
- 알고리즘 - 나간 시간과 들어온 시간간의 차를 구하여 second를 더하고, sum을 더하고, allTime을 빼는 식이다!

```
@objc func pauseWhenBackground(noti: Notification) {
        print(“background")
}

 @objc func willEnterForeground(noti: Notification) {
        print(“Enter")
}
```

12171571 강민상 / FDEE

Xcode

Xcode Swift 프로그래밍

- 1. 백그라운드 구현 - 두가지 함수를 실행시키기 위한 작업

```
var isStop = true
var diffHrs = 0
var diffMins = 0
var diffSecs = 0

NotificationCenter.default.addObserver(self, selector: #selector(pauseWhenBackground(noti:)),
name: UIApplication.didEnterBackgroundNotification, object: nil)
NotificationCenter.default.addObserver(self, selector: #selector(willEnterForeground(noti:)),
name: UIApplication.willEnterForegroundNotification, object: nil)
```

12171571 강민상 / FDEE

Xcode

Xcode Swift 프로그래밍

- 2. 백그라운드 구현 - 화면 나갈시, 화면 들어올시 메소드

```
if(!isStop)
{
     realTime.invalidate()
     timeTrigger = true
     let shared = UserDefaults.standard
     shared.set(Date(), forKey: "savedTime")
}

if(!isStop)
{
     if let savedDate = UserDefaults.standard.object(forKey: "savedTime") as? Date {
          (diffHrs, diffMins, diffSecs) = ViewController.getTimeDifference(startDate:
savedDate)
          refresh(hours: diffHrs, mins: diffMins, secs: diffSecs)
          removeSavedDate()
     }
}
```

Xcode

Xcode Swift 프로그래밍

- 3. 백그라운드 구현 - 시간차를 구하는 함수

```
static func getTimeDifference(startDate: Date) -> (Int, Int, Int) {
    let calendar = Calendar.current
    let components = calendar.dateComponents([.hour, .minute, .second], from: startDate, to: Date())
    return(components.hour!, components.minute!, components.second!)
}
```

12171571 강민상 / FDEE

Xcode

Xcode Swift 프로그래밍

- 4. 백그라운드 구현 - 구해진 시간차로 allTime, sum, second 값을 수정하는 함수

```
    func refresh (hours: Int, mins: Int, secs: Int) {
        let tempSeconds = hours*3600 + mins*60 + secs
        if(second - tempSeconds < 0)
        {
            allTime = allTime - second
            sum = sum + second
            second = 0
        }
        else if(allTime - tempSeconds < 0)
        {
            allTime = 0
            sum = sum + allTime
            second = second - allTime
        }
        else
        {
            allTime = allTime - tempSeconds
            sum = sum + tempSeconds
            second = second - tempSeconds
        }
        startAction()
    }
```

Xcode

Xcode Swift 프로그래밍

- 5. 백그라운드 구현 - 시간텀을 위해 저장한 임시저장값을 제거하는 함수, 시작메소드 생성

```
    func removeSavedDate() {
        if (UserDefaults.standard.object(forKey: "savedTime") as? Date) != nil {
            UserDefaults.standard.removeObject(forKey: "savedTime")
        }
    }

func startAction()
```

12171571 강민상 / FDEE

Xcode

Xcode Swift 프로그래밍 - 도전!

- 빡공률 구현 - 고민해보세요!

```
    func checkPersent()
    {
        if let startTime = UserDefaults.standard.object(forKey: "startTime") as? Date {
            (diffHrs, diffMins, diffSecs) = ViewController.getTimeDifference(startDate: startTime)
            timeForPersent = diffHrs*3600 + diffMins*60 + diffSecs
            print("timeForPersent : " + String(timeForPersent))

            //계산부분
            let per : Double = Double(sum)/Double(timeForPersent)*100
            persentLabel.text = "빡공률 : " + String(format: "%.1f", per) + "%"

            if (per>50.0)
            {
                persentLabel.textColor = UIColor.white
            }
            else
            {
                persentLabel.textColor = TEXT
            }
        }
    }
```

Xcode

Xcode Swift 프로그래밍 - 도전!

- 빡공률 구현 - 알고리즘 간략한 설명

- 타이머 맨처음 시작시 (sum = 0에서 시작할때) 시간을 userDefaults로 저장!
- 타이머가 진행되면서 현재시간과 저장된 시간간의 차로 “실제 지나간 초”계산
- 타이머에 누적된 sum값과 비교를 통해 빡공률을 알 수 있다!
- 빡공률 = 공부한 시간(sum) / 실제 지나간 시간 * 100
- 표시 : persentLabel.text = "빡공률 : " + String(format: "%.1f", per) + "%"

12171571 강민상 / FDEE

Xcode

Xcode Swift 프로그래밍 - 도전!

- 프로그래스 구현 - 관련 유튜브 링크

- 링크 : https://www.youtube.com/watch?v=Qh1Sxict3io

- CircularProgressView.swift 파일 완성시
- StoryBoard에 View를 생성, Outlet으로 연결 (예시 : CircleView)
- Type을 View -> CircularProgressView로 변경

```
@IBOutlet var CircleView: CircularProgressView!
```

Xcode

Xcode Swift 프로그래밍 - 도전!

- 프로그래스 구현 - 사용방법
- 시작점, 도착점을 잘 정해야 한다
- 어플 실행시 예시 - 시작점 : 0.0
- 어플 실행시 예시 - 도착점
- 전체 총 타이머시간(fixedSecond) - 남은 타이머시간(second) / 전체 타이머시간(fixedSecond)

```
        //프로그래스 추가
        CircleView.trackColor = UIColor.darkGray
        CircleView.progressColor = BABYRED!
        fixedSecond = UserDefaults.standard.value(forKey: "second") as? Int ?? 3000

        progressPer = Float(fixedSecond - second) / Float(fixedSecond)
        fromSecond = progressPer
        CircleView.setProgressWithAnimation(duration: 1.0, value: progressPer, from: 0.0)
```

Xcode

Xcode Swift 프로그래밍 - 도전!

- 프로그래스 구현 - 사용방법

- 1초마다 프로그래스 업데이트 - 시작점 : 이전계산값

```
fromSecond = progressPer
```

- 1초마다 프로그래스 업데이트 - 도착점 : 감소된 second로 재 계산된 값

```
//프로그래스 추가!
progressPer = Float(fixedSecond - second) / Float(fixedSecond)
CircleView.setProgressWithAnimation(duration: 0.0, value: progressPer, from: fromSecond)
fromSecond = progressPer
```

Xcode

Xcode Swift 프로그래밍 - 도전!

- 빡공률과 프로그래스 비교 코드가 필요하다면
- 출시된 TiTi 깃허브 링크
- https://github.com/FreeDeveloper97/Project_Timer

종강!!

TMI?

짧막한 느낀점

- 처음으로 진행해본 스터디
- 사람이 없진 않을까, 따라올 수 있을까, 설명을 잘 할수 있을까 등등 걱정 태반
- 잘 따라와주셔서, 과제 잘해주셔서 감사합니다! ^^
- 코드는 구현도 중요하지만 메소드를 활용해서 보기좋게
- 저도 처음부터 완벽하게 딱딱 순서대로 만든게 아니다보니 설명하기 어렵다는점...
- 설명을 위해 다시 코딩하다보니 코드정리를 알게된 느낌

TMI?

짧막한 소개

- 개인 블로그 ☞ https://fdee.tistory.com
- 만들어본 어플 ☞ https://fdee.tistory.com/entry/블로그-시작
- 진행중인 프로젝트 : 타자게임어플 (ios), 미세먼지를 Java (android)

- 좋은 아이디어로 만들고싶은게 있다면 혹시모르니 연락주세요~!

감사합니다!