



AVFoundation +
Dark Mode

애플 스타일 뮤직 앱 제작

Presenter Ye Jin-hee

INDEX

AVFoundation +

Dark Mode

애플 스타일 뮤직 앱 제작

제작개요 01

개발 및 구현 02

영상 및 관리 03

논의 및 고찰 04

01. 제작개요



음악파일에 접근하여 metadata
내용으로 컬렉션 뷰 내용 갱신



AVFoundation, AVPlayer를
사용하여 음악 재생, 정지, seeking구현



다크모드를 구현하여 어두운
환경에서의 사용자경험을 향상

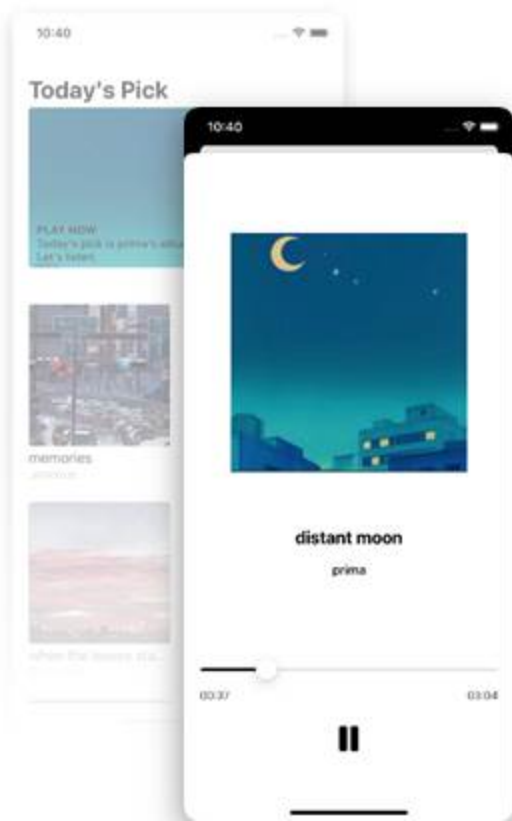
02. 개발 및 구현 - 메인화면

```
1 // HomeViewController.swift -----
2
3 class HomeViewController: UIViewController {
4     // 트랙관리 객체 추가
5     let trackManager: TrackManager = TrackManager()
6     override func viewDidLoad() {
7         super.viewDidLoad()
8     }
9 }
10
11 extension HomeViewController: UICollectionViewDataSource {
12     // 몇개 표시 할까?
13     func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {
14         return trackManager.tracks.count // 트랙메니지에서 가져올것수
15     }
16
17     // 어떻게 표시 할까?
18     func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {
19         guard let cell = collectionView.dequeueReusableCell(withReuseIdentifier: "TrackCollectionViewCell", for: indexPath) as?
20             TrackCollectionViewCell else { return UICollectionViewCell() }
21         let track = trackManager.track(at: indexPath.item)
22         cell.updateUI(item: track)
23         return cell
24     }
25 }
```



1. cell을 TrackCollectionViewcell 형식의 재사용셀로 정의
2. cell의 UI를 트랙정보로 업데이트

02. 개발 및 구현 - 플레이어화면



```
28 extension HomeViewController: UICollectionViewDelegate {
29     // 클릭했을때 어떻게 할까?
30     func collectionView(_ collectionView: UICollectionView, didSelectItemAt indexPath: IndexPath) {
31         // TODO: 곡 클릭시 플레이어뷰 띄우기
32         let playerStoryboard = UIStoryboard.init(name: "Player", bundle: nil)
33         guard let playerVC = playerStoryboard.instantiateViewController(identifier: "PlayerViewController")
34             as? PlayerViewController else { return }
35
36         let item = trackManager.tracks[indexPath.item]
37         playerVC.simplePlayer.replaceCurrentItem(with: item)
38         present(playerVC, animated: true, completion: nil)
39     }
40 }
```

각 셀 클릭 시 Player뷰컨트롤러를 띄움

1. playerVC : PlayerViewController로 형 변환 후 할당
2. 선택한 indexPath의 정보로 playerVC의 현재 아이템을 교체

02. 개발 및 구현 - 플레이어화면

1

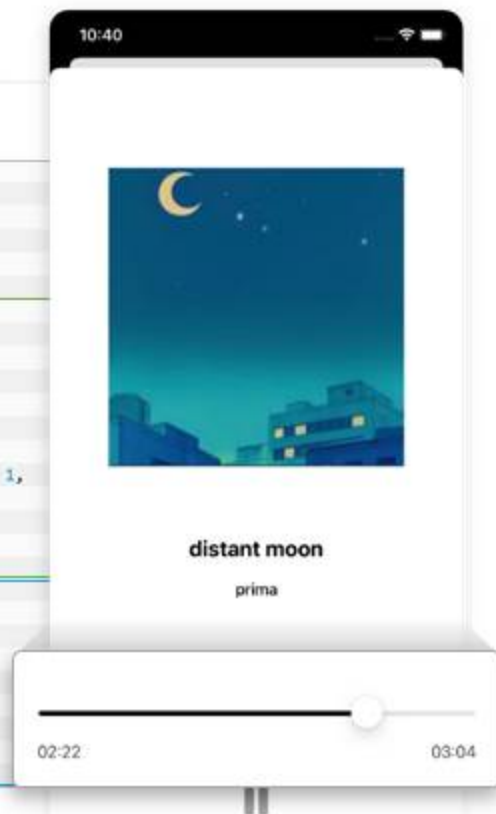
뷰로드 시
play버튼과
재생 시간을
갱신

2

Seeking
구현

- 1 재생하려는 구간
- 2 재생구간 계산
- 3 seek메소드 호출

```
7 class PlayerViewController: UIViewController {
18     let simplePlayer = SimplePlayer.shared // SimplePlayer 만들고 프로퍼티 추가
19
20     var timeObserver: Any?
21     var isSeeking: Bool = false
22
23     override func viewDidLoad() {
24         super.viewDidLoad()
25
26         updatePlayButton()
27         updateTime(time: CMTIME_ZERO)
28
29         // TimeObserver 구현
30         timeObserver = simplePlayer.addPeriodicTimeObserver(forInterval: CMTIME_SECONDS_1,
31             preferredTimescale: 10), queue: DispatchQueue.main, using: { time in
32             self.updateTime(time: time)
33         })
34     }
52     @IBAction func seek(_ sender: UISlider) {
53         // 시퀀스 구현
54         guard let currentItem = simplePlayer.currentItem else { return }
55         1 let position = Double(sender.value) // 0...1 > 0.5
56         2 let seconds = position * currentItem.duration.seconds
57         3 let time = CMTIME_SECONDS(seconds, preferredTimescale: 100)
58         simplePlayer.seek(to: time)
59     }
60 }
```



02. 개발 및 구현 - 플레이어화면

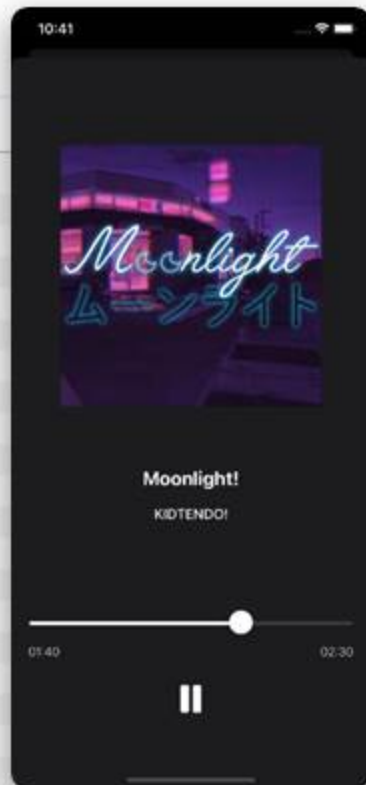
```
61 @IBAction func togglePlayButton(_ sender: UIButton) {
62     // 플레이어 버튼 토글 구현
63     if simplePlayer.isPlaying {
64         simplePlayer.pause()
65     } else {
66         simplePlayer.play()
67     }
68     updatePlayButton()
69 }
109 func updatePlayButton() {
110     // 플레이어 버튼 업데이트 UI작업 > 재생/멈춤
111     if simplePlayer.isPlaying {
112         let configuration = UIImage.SymbolConfiguration(pointSize: 40)
113         let image = UIImage(systemName: "pause.fill", withConfiguration: configuration)
114         playControlButton.setImage(image, for: .normal)
115     } else {
116         let configuration = UIImage.SymbolConfiguration(pointSize: 40)
117         let image = UIImage(systemName: "play.fill", withConfiguration: configuration)
118         playControlButton.setImage(image, for: .normal)
119     }
120 }
121 }
```

1. 곡이 재생중일 때 : 곡 정지 / 2. 곡 정지 상태일 시 : 곡 재생
3. 플레이 컨트롤 버튼 이미지 업데이트



02. 개발 및 구현 - Dark mode

```
1 // DefaultStyle.swift
2 import UIKit
3
4 public enum DefaultStyle {
5     public enum Colors {
6         public static let tint: UIColor = {
7             if #available(iOS 13.0, *) {
8                 return UIColor { traitCollection in
9                     if traitCollection.userInterfaceStyle == .dark {
10                         return .white
11                     } else {
12                         return .black
13                     }
14                 }
15             } else {
16                 return .black
17             }
18         }()
19     }
20 }
```



1. iOS버전이 13.0 이상일 때 다크모드 이용가능
2. userInterfaceStyle별 UIColor 설정 (다크모드면 흰색, 그 외에는 검정)

03. 영상 및 형상관리

슬라이드 쇼 중일 때 이미지가 재생됩니다.

GitHub QR-Code



논의 및 고찰

앱 기능 확장 및 구현

앱을 위젯으로 확장 : 홈 화면에서
위젯으로 사용하여 접근성 향상

system color 외 색상으로 다크모드 대응
되도록 한다면 시각적으로 더 풍부해질 것