

Annaly Rocha

Sept. 11, 2023

Lab 2

COP 4655 - Mobile Apps

Step 3

The screenshot shows the Xcode interface with three main panes:

- Left pane (Project Navigator):** Shows the project structure with files like Track, TracksViewController, DetailViewController, AppDelegate, SceneDelegate, Main, Assets, LaunchScreen, and Info.
- Middle pane (Code Editor):** Displays the code for the Track struct and its mock implementation. The code defines a Track struct with properties like trackName, artistName, artworkURL, and artworkURL00. It also contains a static array of mock tracks for BLACKPINK & Selena Gomez.
- Right pane (Attributes Inspector):** Shows the attributes for a selected UI element (likely a cell). The text field contains "Ice Cream" and "BLACKPINK & Selena Gomez". Other settings include color (Default), font (Body), alignment (Center), and behavior (Enabled).
- Bottom pane (Preview):** Shows a preview of the storyboard. A table view cell displays the song "Ice Cream" by "BLACKPINK & Selena Gomez".

I step up the main view and added the code step 3 requested to add on.

Step 4

```
1 // TrackCell.swift
2 // lab-tunley
3 //
4 //
5 // Created by Annaly on 9/10/23.
6 //
7
8 import UIKit
9 import Nuke
10
11 class TrackCell: UITableViewCell {
12
13     @IBOutlet weak var trackImageView: UIImageView!
14     @IBOutlet weak var trackNameLabel: UILabel!
15     @IBOutlet weak var artistNameLabel: UILabel!
16
17     override func awakeFromNib() {
18         super.awakeFromNib()
19         // Initialization code
20     }
21
22     override func setSelected(_ selected: Bool,
23         animated: Bool) {
24         super.setSelected(selected, animated: animated)
25
26         // Configure the view for the selected state
27     }
28
29     /// Configures the cell's UI for the given track.
30     func configure(with track: Track) {
31         trackNameLabel.text = track.trackName
32         artistNameLabel.text = track.artistName
33
34         // Load image async via Nuke library image
35         // loading helper method
36         Nuke.loadImage(with: track.artworkUrl100, into:
37             trackImageView)
38     }
}
```

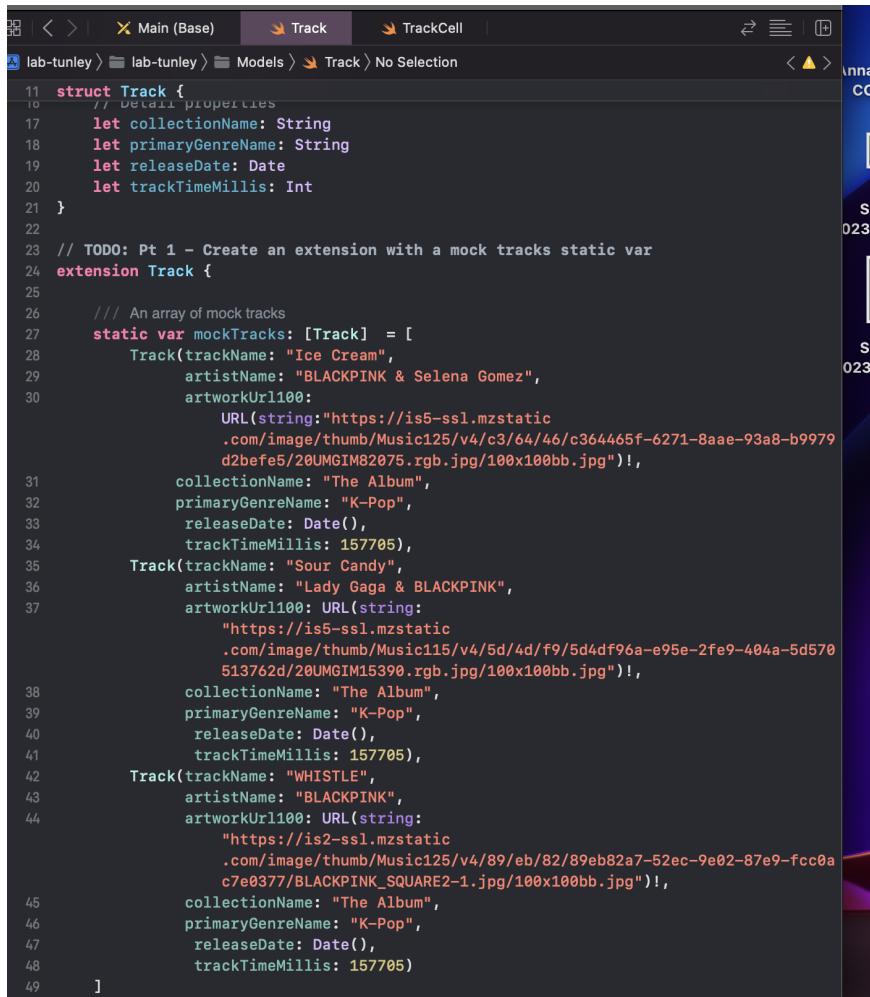
Created a TrackCell file and add the following code and added outlets to be recognized in the code

Step 5

```
| < > ⌂ A...ic > TracksViewController.swift > M tableView(_:cellForRowAt:) 10
11 class TracksViewController: UIViewController, UITableViewDelegate...
12
13     func tableView(_ tableView: UITableView,
14         cellForRowAt indexPath: IndexPath) ->
15             UITableViewCell {
16     // Get a cell with identifier, "TrackCell"
17     // the `dequeueReusableCell(withIdentifier:)` method just returns a generic
18     // UITableViewCell so it's necessary to cast it to our specific custom cell.
19     let cell =
20         tableView
21             .dequeueReusableCell(withIdentifier:
22                 "TrackCell", for: indexPath) as! TrackCell
23
24     // Get the track that corresponds to the table
25     // view row
26     let track = tracks[indexPath.row]
27
28     // Configure the cell with it's associated track
29     cell.configure(with: track)
30
31     // return the cell for display in the table view
32     return cell
33 }
34
35 // TODO: Pt 1 - Add a tracks property
36 var tracks: [Track] = []
37
38 // TODO: Pt 1 - Add table view outlet
39 @IBOutlet weak var tableView: UITableView!
40
41 override func viewDidLoad() {
42     super.viewDidLoad()
43
44     // TODO: Pt 1 - Set tracks property with mock
45     tracks.array
46     tracks = Track.mockTracks
47     print(tracks)
48
49     tableView.dataSource = self
```

We set up the table view by connecting an outlet from the table view in the storyboard to the TracksViewController naming the outlet tableView.

Step 6



```
11 struct Track {
12     // Detail properties
13     let collectionName: String
14     let primaryGenreName: String
15     let releaseDate: Date
16     let trackTimeMillis: Int
17 }
18
19 // TODO: Pt 1 - Create an extension with a mock tracks static var
20 extension Track {
21
22     /// An array of mock tracks
23     static var mockTracks: [Track] = [
24         Track(trackName: "Ice Cream",
25               artistName: "BLACKPINK & Selena Gomez",
26               artworkUrl100:
27                   URL(string:"https://is5-ssl.mzstatic
28                     .com/image/thumb/Music125/v4/c3/64/46/c364465f-6271-8aae-93a8-b9979
29                     d2bef5/20UMGIM82075.rgb.jpg/100x100bb.jpg")!,
30               collectionName: "The Album",
31               primaryGenreName: "K-Pop",
32               releaseDate: Date(),
33               trackTimeMillis: 157705),
34         Track(trackName: "Sour Candy",
35               artistName: "Lady Gaga & BLACKPINK",
36               artworkUrl100: URL(string:
37                   "https://is5-ssl.mzstatic
38                     .com/image/thumb/Music115/v4/5d/4d/f9/5d4df96a-e95e-2fe9-404a-5d570
39                     513762d/20UMGIM15390.rgb.jpg/100x100bb.jpg")!,
40               collectionName: "The Album",
41               primaryGenreName: "K-Pop",
42               releaseDate: Date(),
43               trackTimeMillis: 157705),
44         Track(trackName: "WHISTLE",
45               artistName: "BLACKPINK",
46               artworkUrl100: URL(string:
47                   "https://is2-ssl.mzstatic
48                     .com/image/thumb/Music125/v4/89/eb/82/89eb02a7-52ec-9e02-87e9-fcc0a
49                     c7e0377/BLACKPINK_SQUARE2-1.jpg/100x100bb.jpg")!,
50               collectionName: "The Album",
51               primaryGenreName: "K-Pop",
52               releaseDate: Date(),
53               trackTimeMillis: 157705)
54     ]
55 }
```

We added mock tracks for when you navigate to the other page it will generate the appropriate information about the track selected.

Step 7

The following Just display the codes added and the phone display of how everything put together will run. Unfortunately, my constraints seem to make the cell squeezed but for the most part the app seems to be running fine besides the error there.

```

    class TracksViewController: UIViewController, UITableViewDataSource {
        override func viewDidLoad() {
            super.viewDidLoad()
            // TODO: Pt 1 - Pass the selected track to the detail view controller
            // Set the track on the detail view controller
            if let cell = sender as? UITableViewCell,
                let indexPath = tableView.indexPath(for: cell),
                let detailViewController = segue.destination as? DetailViewController {
                let track = tracks[indexPath.row]
                detailViewController.track = track
            }
        }
        // TODO: Pt 1 - Add table view data source methods
        override func viewDidAppear(_ animated: Bool) {
            super.viewDidAppear(animated)
            // Get the index path for the current selected table view row (if exists)
            if let indexPath = tableView.indexPathForSelectedRow {
                // Select the row at the corresponding index path
                tableView.deselectRow(at: indexPath, animated: true)
            }
        }
    }

```

```

    class DetailViewController: UIViewController {
        var track: Track!
        override func viewDidLoad() {
            super.viewDidLoad()
            // Load the image located at the 'artworkUrl100' URL and set it on the image view.
            Nuke.loadImage(with: track.artworkUrl100, into: trackImageView)
            // Set labels with the associated track values.
            trackNameLabel.text = track.trackName
            artistLabel.text = track.artistName
            albumLabel.text = track.collectionName
            genreLabel.text = track.primaryGenreName
            // Create a date formatter to style our date and convert it to a string
            let dateFormatter = DateFormatter()
            dateFormatter.dateFormat = "medium"
            releaseDateLabel.text = dateFormatter.string(from: track.releaseDate)
            // Use helper method to convert milliseconds into 'mm:ss' string format
            durationLabel.text = formattedTrackDuration(with: track.trackTimeMillis)
        }
        // TODO: Pt 1 - Configure the UI elements with the passed in track
    }

```

