



iOS Training Program: Week 2 Mini-Project

Pokédex

OVERVIEW

The Pokémon franchise started off in its first generation with its initial release of Pocket Monsters Aka and Midori ("Red" and "Green", respectively) for the Game Boy in Japan on February 27, 1996. When these games proved extremely popular, an enhanced Pokémon Ao ("Blue") version was released sometime after, and the Ao version was reprogrammed as Pokémon Red and Blue for international release. The games launched in the United States on September 30, 1998 and have been an international smash-hit ever since.

The number of Pokémon has grown dramatically, and it can be hard to keep track of all of them. This week we'll be building a simple Pokédex app for people to look up statistics and information for the over 800 Pokémon currently in existence.

SPEC

For this project, you are again required to handle the layout programmatically (only using Storyboard to set the navigation between ViewControllers but not actually populating each ViewController with subviews). If you are unsure about what this means, ask the instructor! Your app should be easily navigable – use navigation controllers to allow people to go back freely.

We will be very picky about design choices for this project and all projects going forward. Make sure the app looks polished!

Code quality is just as important as functionality! Avoid redundancy and make your code modular wherever possible. You should be consistent with the style guidelines for Swift (see here: <https://github.com/raywenderlich/swift-style-guide>) - make your variable/function names clear and comment code when things are potentially confusing. Learn the content in this guide! We use it across all projects in MDB.

You will be provided a starter Xcode project with the following files:

- A JSON file containing the raw data for all of the Pokemon, including their name and statistics.
- A Pokemon object class to represent each Pokemon and stores their attributes.
- A static PokemonGenerator class which parses through the JSON file and creates an array of all Pokemon. You can get this array by calling `PokemonGenerator.getPokemonArray()`.

Do not edit the JSON or PokemonGenerator files! They are provided to make data retrieval easier for you. You will also not need to interact directly with the JSON file – just the Pokemon class for accessing attributes and PokemonGenerator for getting the initial array. You may feel free to add any utility functions to the Pokemon class (though not necessary), but do not delete any of the existing code.

There are 3 parts to this project. **Read through all parts before beginning!** They affect each other.

Part 1

- Opens to a search page where users can search either by category or for specific Pokemon.
- Categories include type (“Grass”, “Water”, etc.) as well as minimum attack points, minimum defense points, and minimum health points
 - ‘Type’ categories should be explicitly presented as different options that the user can select (do not allow the user to specify types); also, the user can select multiple types before querying for the list.
 - See the Pokemon class documentation for a list of all types. There are a lot of them, so think carefully about how you want to present them on the screen. You may need to use a ScrollView on the search screen if there’s too much information to fit on the window.
 - ‘Point’ categories should include a textfield where users specify a cutoff number (for example, all Pokemon with attack points above 100).
- Individual Pokemon can be searched by name or by number.
- Users should also be able to randomly get a list of 20 different Pokemon.

Part 2

- If the user chooses randomly or by category, the app should move to a screen that displays a list of all the Pokemon that match the criteria.
- This list should be able to toggle between a list view (using a TableView) or a grid view (using a CollectionView) – you’ll want to use a segmented control to switch between these.
- In the list view, each cell should display a small picture, as well as the name and number of the Pokemon. A single search can include as many as 800 Pokemon, so don’t make your cells too big!
- In the grid view, Pokemon should be displayed in square-shaped cells filled with that Pokemon’s image and with its name overlaid on the bottom corner of the image.

Part 3

- If the user clicks on one of the Pokemon from the list OR searches for a specific Pokemon in the search page, they should go to a detail screen that displays the Pokemon's picture, name and number, and all available statistics. This is essentially a "profile" for the Pokemon, so make it look like a profile page.
- Add a button "Add to Favorites" which LOCALLY stores that Pokemon in an array of favorites. You'll also need to add an additional tab called "My Favorites" to the app (do this using a tab bar controller) where the favorite Pokemon are displayed in a TableView (don't worry about toggling views here or making the cells clickable, but the cells should look similar to the existing list view).
- Add another button called "Search on the Web" which takes the user to a full WebView displaying the Google Search result for that Pokemon (hint: the URL for this would be <https://google.com/search?q=POKEMONNAME>)
- **Note:** Some of the urls for Pokemon are invalid. You need to check the urls before you use them. Otherwise your app will crash! Remember to do this. We will be assessing it.

Last but not least, add a README outlining the purpose of the app and which features are included.

SUBMISSION

Submit your application using <https://goo.gl/forms/gEtQhulCeXY4v3rM2>. Remember that these projects are due on 2/16 at 11:59 pm. Even if you have not finished the project, submit before the deadline!