COMP228/327 Assignment 2   w/c 14th December 2020

Developing an "Artworks on Campus" App.

Your Task:

You will design and develop an application written in Swift 5 using UIkit and Storyboard for iPhone 11. The application will enable you to locate artworks on campus relative to the user's current location.

In order to do this you will need to retrieve data from a web service regarding the location of, and information about, artwork on campus.

https://cgi.csc.liv.ac.uk/~phil/Teaching/COMP228/artworksOnCampus/data.php?class=artworks&lastModified=2020-12-13

Note that:
- The **lastModifed** parameter is optional and may be used to retrieve only data that has changed since the date provided.
- Images of the artworks, referenced in the JSON data are located at the following base URL:

https://cgi.csc.liv.ac.uk/~phil/Teaching/COMP228/artwork_images/

(Note: use secure URLs, otherwise your app will not load the data or images).

Your application is required to have the following basic features **(worth 75%)**:

1. The user is initially presented with a map centred on their current location and at a reasonable level of zoom so that nearby roads etc. can be seen clearly. You may assume that the user is currently in the Ashton Building (a location file is available for Xcode to simulate the location of the Ashton Building). (latitude: 53.406566, longitude: -2.966531). **(worth 30%)**

2. The map contains a number of annotation marks indicating the location of nearby artworks. If a location (e.g. a building) has multiple artworks then these should be represented by a single annotation on the map. **(worth 5%)**

3. In portrait view, a table below the map contains a list of artworks, grouped by building and ordered by distance from the user's current location. A location file featuring a "walk" around campus is available and can be used to test this feature **(worth 25%)**

4. Tapping on an annotation displays an image and information either about a specific artwork, or else sensibly handles the fact that the annotation represents multiple artworks. One way to do the latter is to present a list of artworks available within the building, selecting any of which then displays an image and information about the selected artwork. Feel free to develop your own UI if you have an alternative that you think is better. **(worth 15%)**

Note that the artwork should be displayed on the map or in the table only If the value of the *enabled* attribute is "1". If it has any other value, that artwork should be ignored.

The remaining **20%** of the marks may be obtained by implementing useful features such as:

1. Caching the artwork information (in Core Data). **(worth 10%)**

2. Synchronising the app on startup, checking to see if new or modified data is available from the web service, using the lastModified parameter. **(worth 10%)**

Please ensure that your code is appropriately commented and that meaningful class, variable and constant names are used **(worth 5%).**

If you use any additional images or other materials, ensure that these are copied into the project – not just referenced somewhere else in your filestore. The zipped folder that you submit should include everything required to compile and run your App.

Important - Please note:
Do not use any third-party frameworks in your App (e.g. Alamofire). Use Apple standard frameworks ONLY (i.e. only frameworks provided with Xcode 12). Use of third-party frameworks will involve a **penalty of 15%.**

What to Submit
Your completed project should be zipped up and submitted via the online submission system:
 https://sam.csc.liv.ac.uk/COMP/Submissions.pl

(In the Finder, right click the icon for the folder containing the project file and folder and choose "Compress")
Also submit a short document (maximum of 1-2 sides of A4) documenting how to use your app and any notable features or limitations.

Deadline for submission: Monday January 11th at 5pm.

Reminder: This is the second of two assignments, each of which is worth 15% of the total mark for COMP228. Your portfolio of lab work will be worth another 10%.
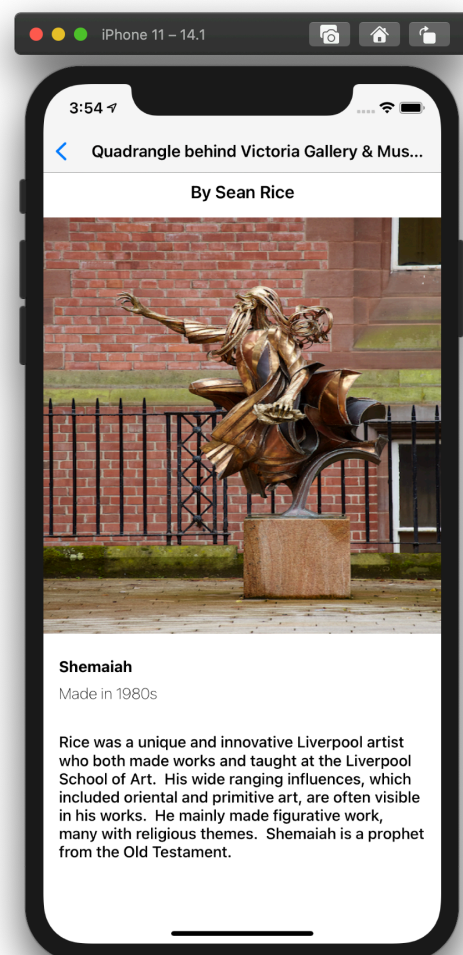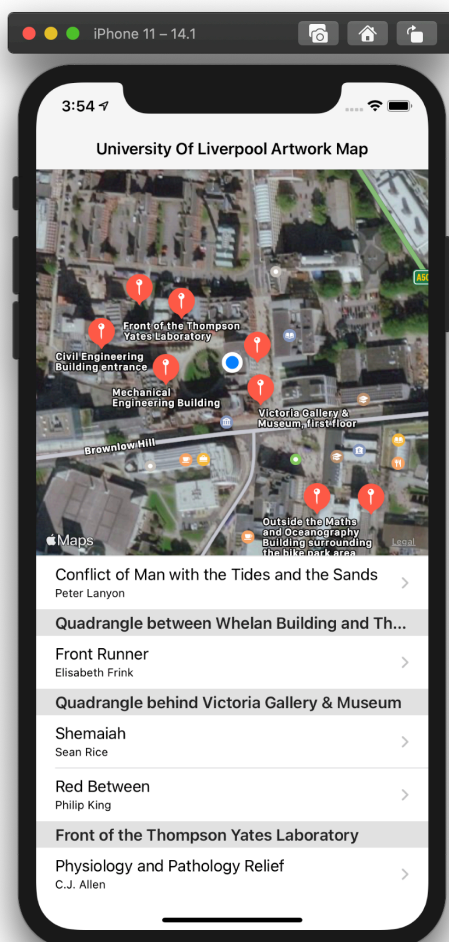
Hints:
The JSON data has the following structure:

```json
{
    "artworks": [
        {
            "id": "62",
            "title": "Square With Two Circles",
            "artist": "Barbara Hepworth",
            "yearOfWork": "1964",
            "type": "sculpture",
            "Information": "Dame Jocelyn Barbara Hepworth DBE was an English artist and
sculptor.  Her work exemplifies Modernism and in particular modern sculpture.  She was
one of the few female artists of her generation to achieve international prominence.
This sculpture is made of bronze with a concrete base.  Hepworth was influenced by the
monolithic power of stone monuments and uses the cut out circles as a device to open up
the solid mass of the structure.",
            "lat": "53.405944",
            "long": "-2.96237",
            "location": "Crown Place",
            "locationNotes": "Crown Place Halls of Residence",
            "fileName": "Square with Two Circles by Hepworth.png",
            "lastModified": "2020-12-13 16:14:13",
            "enabled": "1"
        }
    ]
}
```

Which leads to the following decodable Swift structures:

```swift
struct Artwork: Decodable {

    let id: String
    let title: String
    let artist: String
    let yearOfWork: String?
    let type: String?
    let Information: String?
    var lat: String
    var long: String
    var location: String
    let locationNotes: String?
    let fileName: String?
    let lastModified: String
    let enabled: String?
}

struct AllArtworks: Decodable {
    let artworks: [Artwork]
}
```