



iOS development workshop

iOS 13 - Swift 5

by hyokil.kim@epitech.eu, maxime.gernath@epitech.eu clement.dubois@epitech.eu

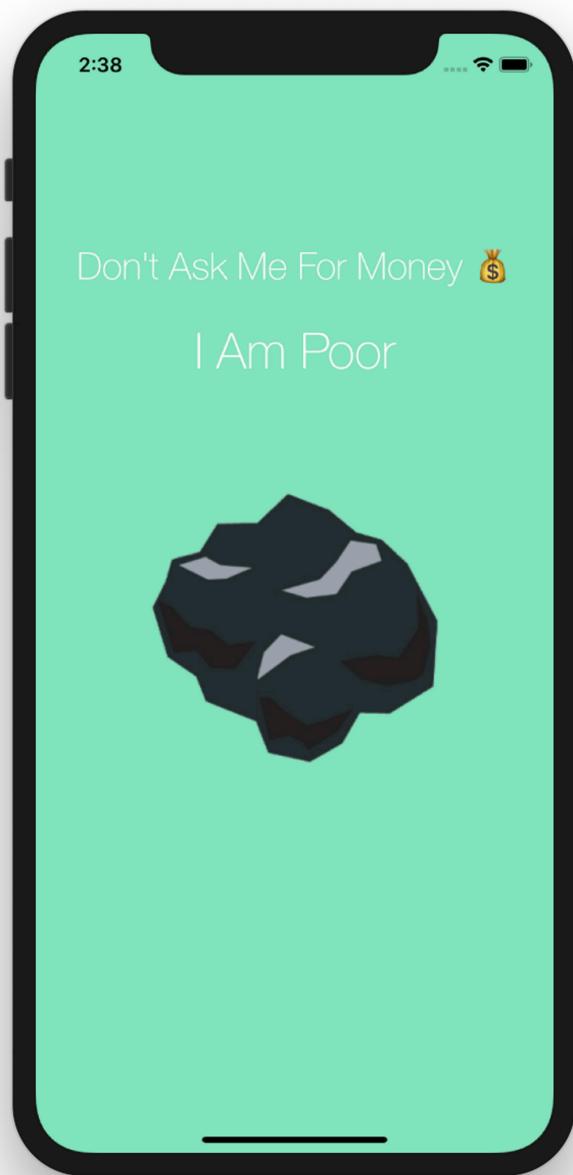


What You Will Create

An useful application . . .

You will create an 'I Am Poor' application. This app will proudly display a lump of coal and the statement "I Am Poor"; the perfect app for when you get asked for money on the streets.

The end result will look something like this. Although there's endless scope for customisation. So hopefully, your app will look even better!





Task 1 – Create a Xcode project

Let's create!

Create a new Xcode project called **I Am Poor** or any other name you would like to give your app.



Xcode autosaves all your progress every few seconds.

This is what you should end up with.

The screenshot shows the Xcode interface with a new project named "I Am Poor". The project structure on the left includes files like AppDelegate.swift, SceneDelegate.swift, ViewController.swift, Main.storyboard, Assets.xcassets, LaunchScreen.storyboard, and Info.plist. The main area displays the General settings for the target "I Am Poor". Under "Identity", the Display Name is set to "I Am Poor" and the Bundle Identifier is "com.angelayu.I-Am-Poor". Under "Deployment Info", the Target is set to "Device" and "iOS 13.0" is selected. The "Main Interface" is set to "Main". Under "App Icons and Launch Images", the App Icons Source is "AppIcon" and the Launch Screen File is "LaunchScreen". The right side of the screen shows the "Identity and Type" sidebar with details like Project Format (Xcode 9.3-compatible), Organization (Angela Yu), and Class Prefix. Text Settings are also visible, including Indent Using (Spaces) and Wrap lines checked.



Task2 – Add a Label element

Some UI 😊

- Using the **Object Library**, drag and drop a **Label** onto the canvas in the **Main.storyboard**
- Set the Label text as “**I Am Poor**” or anything else you wish.
- Change the Label’s **font** and **font size** and **colour**.
- Resize to make the label fit the text. Position the label anywhere you see fit.

You might choose a different font or colour but this is an example of what you can aim for at this stage:



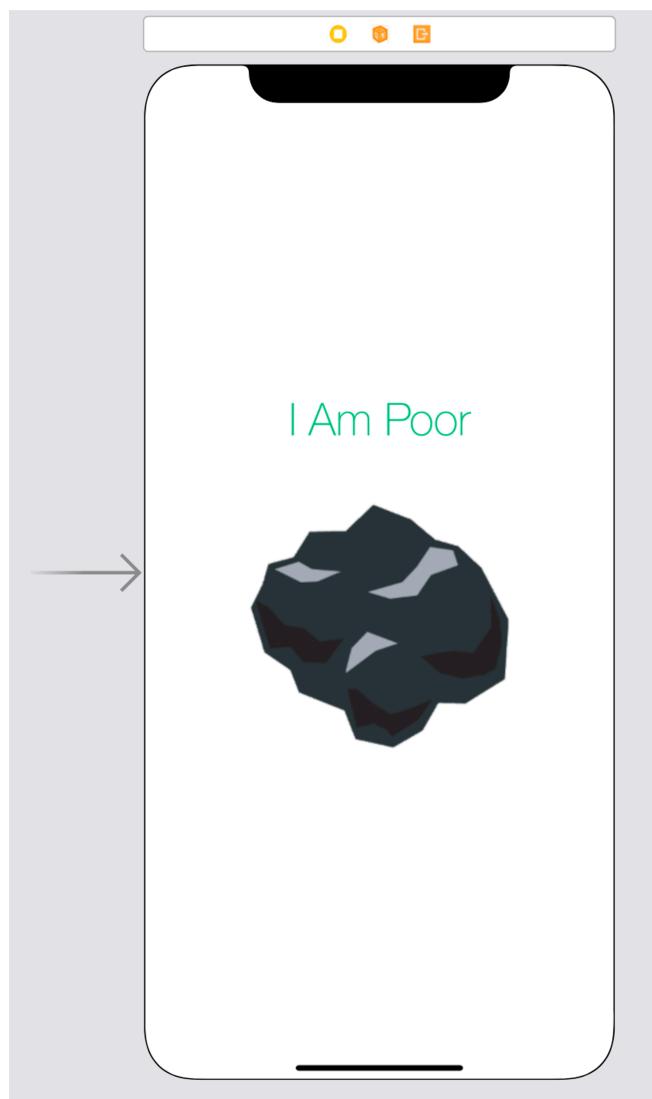


Task3 – Add an Image View

Some UI again 😊

- Again using the Object Library, find an **Image View** and drop it onto the canvas.
- The Image View needs to display an image. Feel free to customise the size and positioning of the Image View.

This is your chance to show off your design skills!



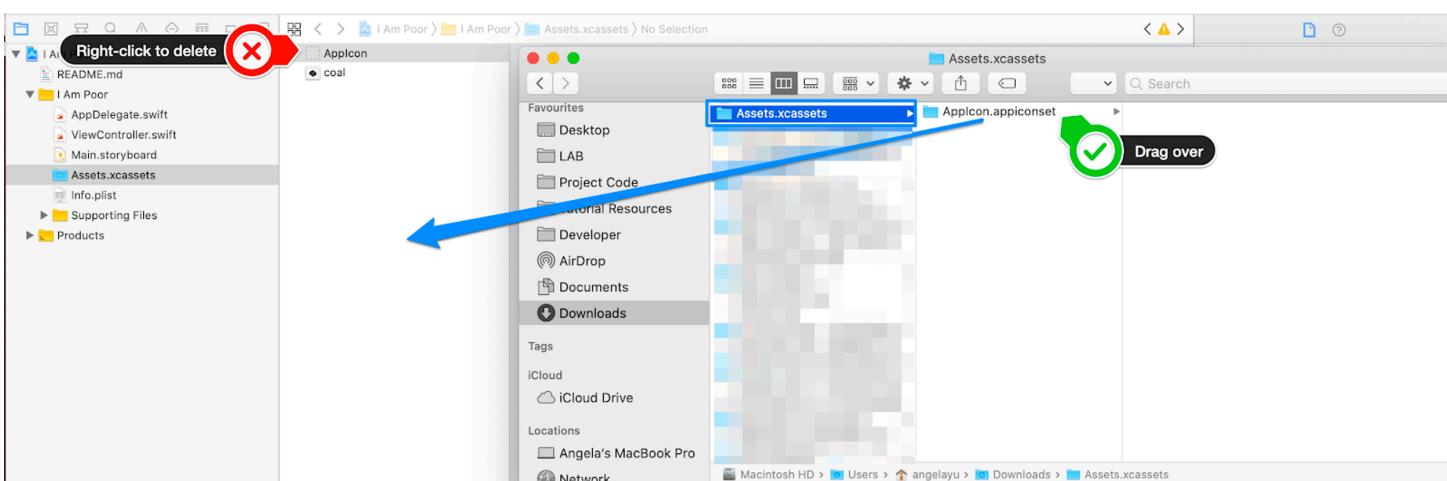


Task4 – Add an App Icon

Some UI again 😊

- use your own images and generate all the required sizes using [appicon.co](#).

Once you've downloaded and extracted the images from appicon, you can match up the sizes or you can simply replace the App Icon folder.



GOAL

When you run the app in the simulator, you should be able to see an app icon on the Home screen.



Task5 – Run your app

Let's run this! 😊

- Feel free to make any other modifications or customisations. e.g. You can add a background colour or some more text/images. It's your app, design it to your heart's content!
- If you have a connected Apple device, you can run the app on your iPhone. If not, you can run it on the Simulator. Remember, the keyboard shortcut to run your app is **command + R**.

Task6 – Change your image !

Some code logic please! 😊

- Add some images to your assets
- Add two buttons, previous and next, in order to change the displayed image.



Check how we set the image of an ImageView programmatically.



Task 7 – Show off your work !

We hope you enjoyed that challenge and you're now familiar with all of Xcode's tiny buttons in Interface Builder.

We really encourage you to take the opportunity and fully customise the design of your app.