Luke and Michael

Programming 3

Mobile CS

Design Rational

Our design rational was to continue following good practices learned earlier in the term. We continued delegating tasks and code to fragments rather than the Main Activity. We also made sure to do planning before tackling problems. The best example in this project was creating the diagram for JSON objects versus actual model classes, to plan out what the code will look like before writing it. Finally, we continued to write functions and classes to divide up code and tried our best to avoid having too long functions and maintain to the MVC design pattern.

Feature Screenshots

Part 1

Below is the screenshot of the relative directory of the drawable resources.

Graphical user interface, application

Description automatically generated

Below is the screenshot of the finalized layout design, found in the Layout Design Editor.

Graphical user interface

Description automatically generated

Below is the screenshot from the Device File Explorer that shows that the full-sized images are saved correctly.

Graphical user interface, text, application

Description automatically generated

Finally, below is the screenshot of the game fragment displaying the two team profiles properly.

A screenshot of a phone

Description automatically generated with medium confidence

Part 2

Below is the Logcat of the response received. A full text version is also in the root directory of the submission.

Text

Description automatically generated

Below is are two screenshots showing the weather information for Worcester, one showing it in ugly JSON and the other showing it in a pretty format.

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

For the data classes that we chose to implement, we made sure to make them follow the actual JSON object as closely as possible. Below is a diagram showing the classes compared to the JSON object.

Diagram

Description automatically generated

Since the only information that is needed for this use-case is temperature (the city name is given already, hard-coded at this point in the project and later gained via location services), only two classes are needed. The main class is what stores information on data like temperature, so that is why that class exists in the code.

What we have learned

In this assignment, we have gotten ourselves more familiar with how to use implicit intents to add new features to our existing application. Specifically, working with third-party apps (such as the camera app) was very useful. We also got familiar with Android’s permission model via FileProvider, for data sharing. We also learned about how to use Retrofit and Gson libraries to work with third-party network systems, to request for a given city and receive its weather information via the api. Finally, working with location data and setting that up was a new experience for both of us. Overall, this project felt akin to a lot of software jobs, where instead of making a new app you extend the features of a preexisting product, thus it was an invaluable programming assignment.