Tae Hyun Je

CS 4518: Mobile and Ubiquitous Computing

Project 3: A More Feature-Rich BasketBall Application

9/15/20

Table of Contents

Design Rationale

This section will go over what the UI looks like and the reasoning behind it.

UI Design

Not much has been changed since project 2 for the initial home screen, MainActivity. The only difference is with the new activity layout and the RecycleViewer. Refer to “Feature Screenshots” for images on them.

Reasoning

The reasoning for the designs was for the sake of simplicity. A cluttered UI can quickly be unappealing to the user and could even throw them off. Most of these additions were based on the example images provided by the homework assignment.

Feature Screenshots

For the first part of the assignment, we had to create another Activity that passed data to and from the original Activity. Using the “Salvar” button, the user is taken to a screen that displays the game score they want to save (see Figure 1).

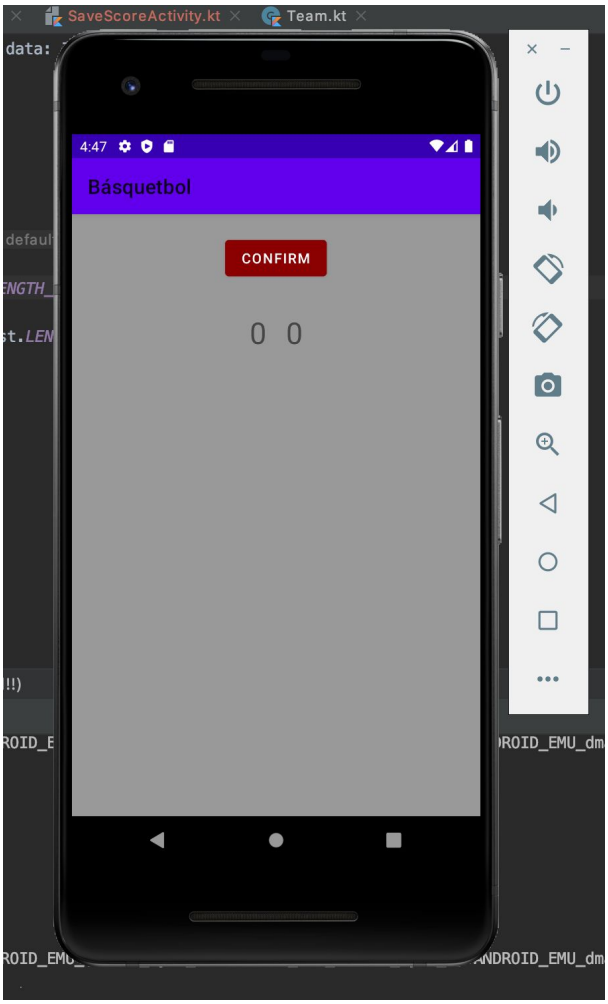

Figure 1. 2nd Activity View

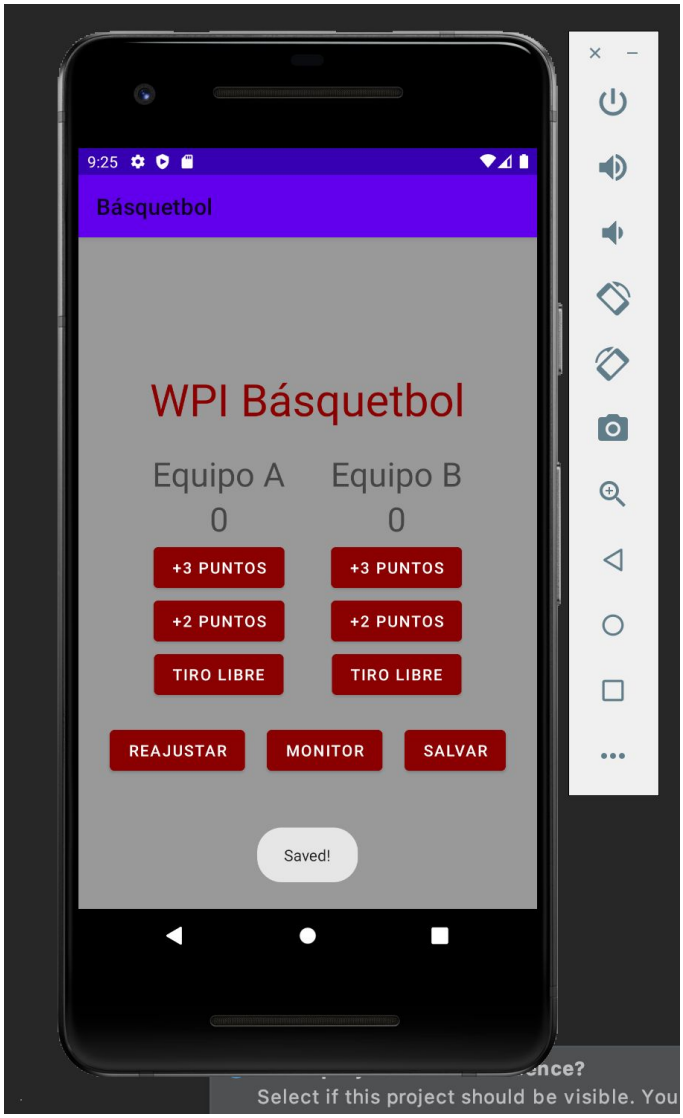

Figure 2. Confirming Data Share with Intent

Furthermore, to showcase that data can also be passed back from the 2nd activity, I added a Toast display. If the user presses the button to save, they will see a message save “Saved”. If they do not, they will not see anything. This is also to indicate that their game has been saved.

Since the 2nd part was just to refactor the code to use fragments, I do not have any pictures, please refer to the code.

The 3rd part of the assignment had to do with the RecyclerView. This was used to display a list of 100 random games. I created an xml called list_item_team that was used with RecyclerView to display all the data (see Figure 3).

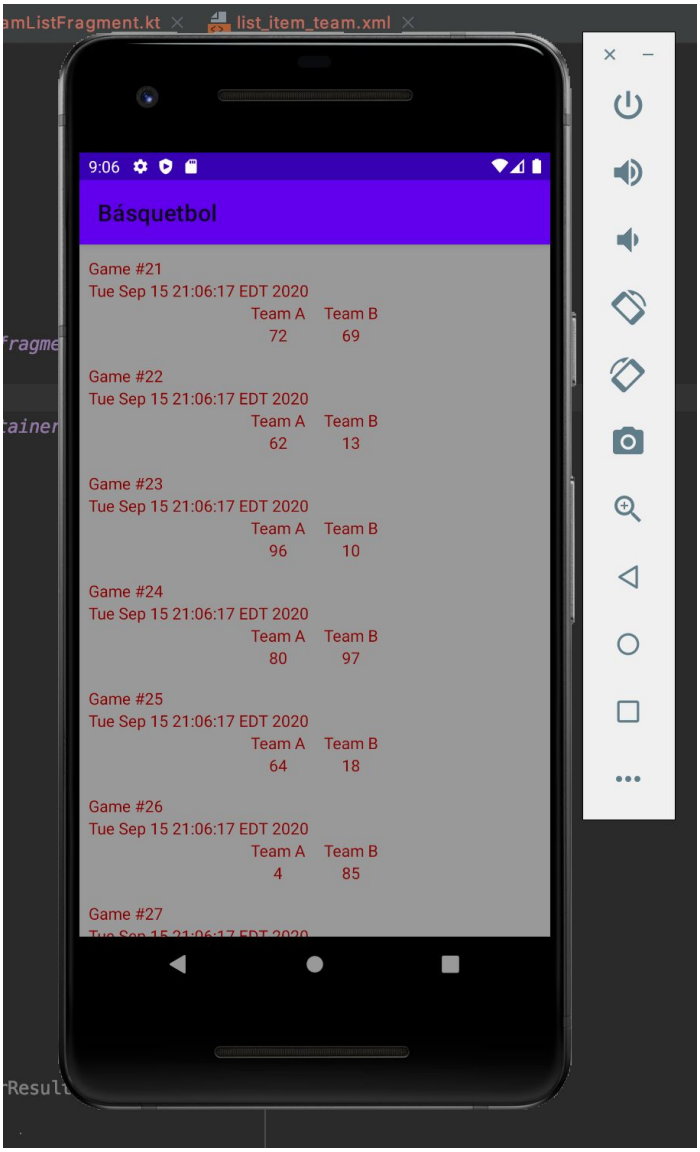

Figure 3. RecyclerView

Reflection

This section will discuss the educational knowledge that was obtained upon completing this project.

Logcat

Like with the previous assignment, I used Logcat to display the sequences taken by the Intent activity between the 1st and 2nd Activities (see Figure 5). It sequentially showed the progress for the important Intent functions: startActivityForResult(), newIntent(), setResult(), and OnActivityResult().

File - unknown

```
1 2020-09-15 17:30:40.343 12474-12474/com.android.
  basketballcounter D/TAGSAVECHECK: onClick() for SAVE is
  called, startActivityForResult()
2 2020-09-15 17:30:40.346 12474-12474/com.android.
  basketballcounter D/TAGSAVECHECK: newIntent(), putExtra()
  has been called
3 2020-09-15 17:30:42.829 12474-12474/com.android.
  basketballcounter D/TAGSAVECHECK: setResult() is called
4 2020-09-15 17:30:44.631 12474-12474/com.android.
  basketballcounter D/TAGSAVECHECK: OnActivityResult() has
  been called
5
```

Figure 5. Logcat for Intent

Sequence Diagram

A sequence diagram was also made prior to capturing the log with Logcat (see Figure 6). It clearly represents the flow of the application when passing data to the 2nd Activity and back, even including how the Android OS is an integral part to it.

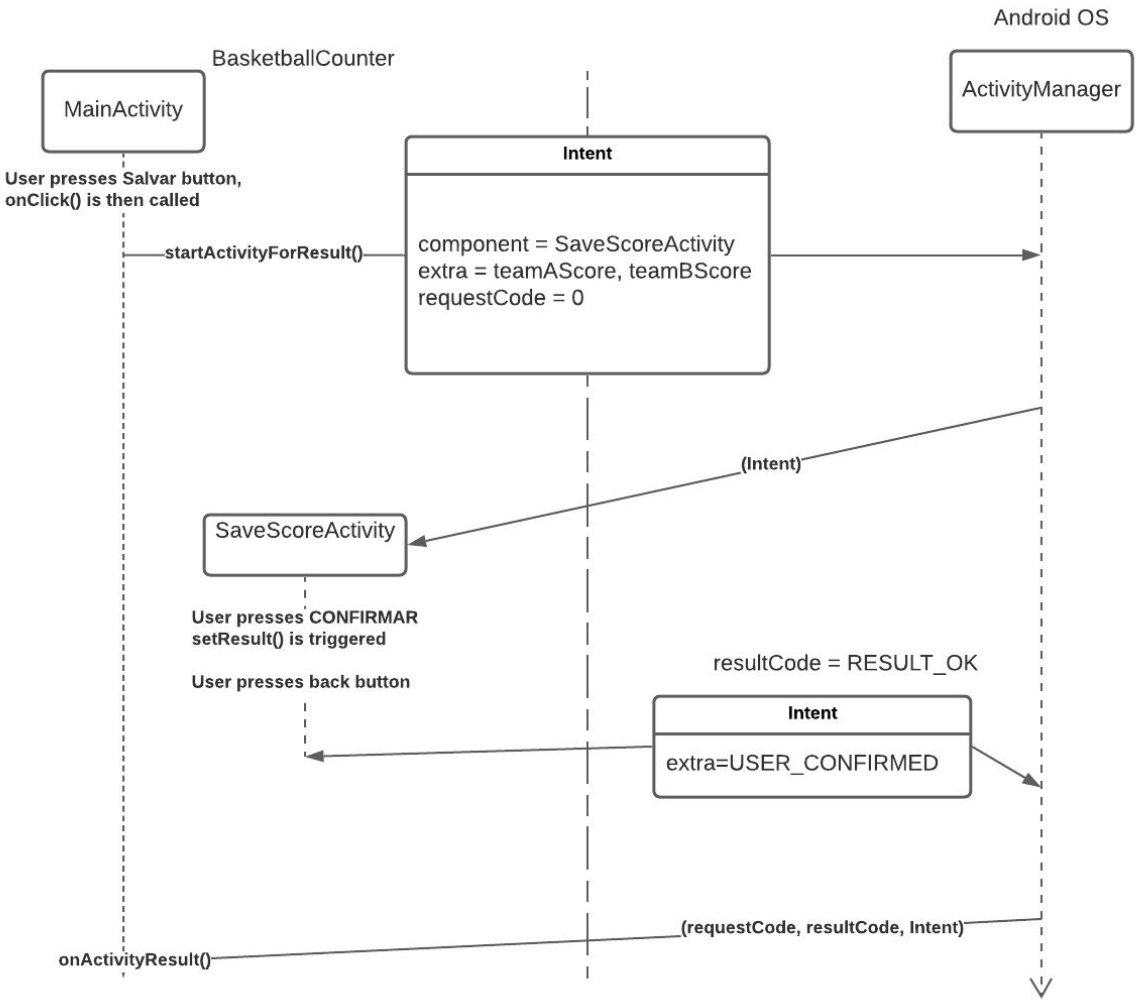

Figure 6. Sequence Diagram for Intent

Conclusion

In conclusion, this project was pretty challenging. While I did not exactly add more features to this app, I was able to test two different system designs using Intent and Fragments. I think that the most challenging part is the different methods to call and override in order to properly pass data and create objects for the views through companion objects. It is not particularly hard, but the lack of experience makes it difficult to quickly get accustomed to. For example, it is much easier to display a list with React Native, it just requires a Flatlist and sharing it between views. There is more to it, but React does a lot of the work in the background. It is essentially the same but I just need more experience with the implementation.