

# Final Project Discussion

Dongyan Li 1370967

---

## API features and classes I have used in the final project:

- 2D graphics drawing:

I used it to draw the brewing curve for the brew screen to help user visualize the brewing progress. For me, this is the most time consuming part of my project.

At first, I want to use the “extends View” method by instruct the UI thread to draw the graphic. Also, I want to make the graphic pause for 1 sec since this is the nature way of representing the brewing progress. However, only after I found that I can’t pause the thread in the middle of the “for loop” and let the handler postDelay since the onDraw method must completely finished for each cycle.

After that, I remember there’s another way of doing that: extending SurfaceView and let second thread to do that. However, there comes another problem: I can’t let this thread to do the updating of other views because only the UI thread can do that.

Then, I went back to the first method: extends View and implement a custom callback method to update other view related to the change of the brewing curve.

- Parcelable for List:

In order to pass the data among the activities, I implemented the Parcelable interface for my custom objects. To be specifically: pass the brewing intervals for the 2D graphic drawings. When I implementing the code, I always get weird numbers that I didn’t pass to the intent. I spend quite a few hours debugging that only to find you need use the same sequence of variable in the field to write to intent as to read from the intent, otherwise, the number is messed up.

- Custom callback

The professor mentioned during the class but for whatever reason I didn’t get through my mind. Only later I need to use this feature I have to spend some time search online to find the answer. I used it to make the 2D drawing class observable and the parent activity could use the update to update other view within the activity.

Discuss any limitations of your app and how do you think the app can be improved:

- The custom view use the main UI thread to update the view every 1 second. I don't think it will impact too much of the performance, however, I did implement the SurfaceView method but I got every confused about the multi-threading and lock mechanism. I used the debug tool to see how precise the UI thread updating the view, the difference is within 100 millis. I think if I implement the surfaceview, the precision could be better.
- The UI part of the app could be better polished and the drawing of brew curve view should get more "modern" look than the more "primitive" look I made.
- The cupping screen need more functions. For example: save the configures to the Cloud. Add an instruction for the coffee flavors.

Discuss some of the biggest challenges you have encountered during the development

- For me, the biggest challenges is the code execution sequence and the scope of the variables. For my custom view, there was a bug that one of my functions can't get the view size, eg:width and height, they are always zero. In the API I know the custom view will call the onSizeChanged every time the view is initiated. So, I assume that any other functions I created and called will get the width and height information. However, I failed everytime. I was every frustrated when I encountered that problem at first. I even thought my machine is broken! But after discovering the problem had me realized you need to calm down and think carefully where the problem is. Panic won't save you any time.I have a sense that it comes form the order of execution of the functions so I went toward that direction. In order to debug that, I put every function a Log.d tag and the name of the function to see the sequence of the execution. To my surprise, my own function that call in the main activity got called before the onSizeChange!

Overall experience as an Android developer

- So far, I love programing as an Android developer! You feel so accomplished when you get everything going on that small screen. Programming for real device is unlike programming in the computer. You can actually make something practical and life easier by making mobile apps. The Internet thing is real happening now.

Discuss any limitations of the Android SDK and Android Studio you have encountered during the final project, as well as during the programming assignments.

- I think Android SDK is well documented and fair easy to use and Android Studio too, is fast and stable most time. Sometime, the Android Studio will freeze and not responding when I editing the XML view file. Also, I keep get null pointer error from the rendering of the view design but the trace are from Android Studio itself but my code.