Dylan Dunham
CEG4110
HW2
10/11/2018

Main Features:

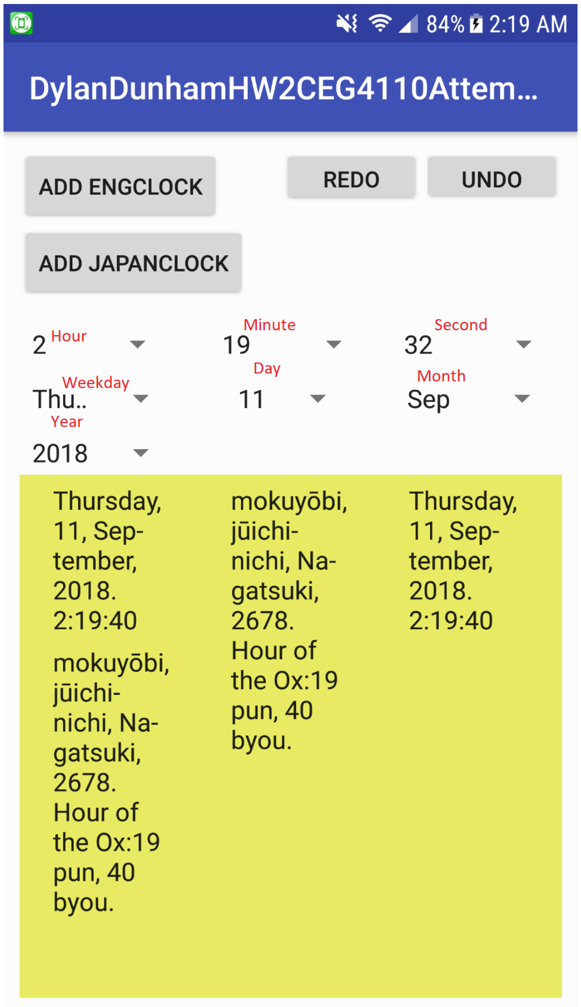
My application displays many traditional western clocks in English and traditional Japanese ways of telling time. By tapping the “ADD ENGCLOCK” button, a western clock is added. By tapping “ADD JAPANCLOCK” a Japanese clock is added. All the clocks added can be changed at once by editing the number in the drop down. From right to left, top to bottom: Hours, Minutes, Seconds, Weekday, Day, Month, Year. By tapping the “UNDO” button the last edit to the clocks is undone. By tapping the “REDO” button the last undone change is redone.

About the Japanese Clock:

The English clock is self explanatory but not so with the Japanese clock. It displays the same information but in traditional Japanese time telling methods when applicable. This includes the hour, month, day, weekday, and year. Minutes and seconds didn’t exist in Edo Japan so the modern Japanese terms are substituted here. The Japanese of course had their own words for each month, day of month, and weekday, but where this clock really differs from the English clock is the hour and year. The Japanese had (usually) 12 hour days that would count down from 9 to 4, then starting at 9 again. Numbers are excluded due to them being reserved by buddhist monks praying time. They also counted down due to the main way to tell time at the time, burning incense, which would decrease throughout the day. Each “hour” (2 western hours) also had its own zodiac animal assigned to it which is represented in the clock here. Though, Japanese hours and how they correspond to western hours changes seasonally, so for simplicity they 24 hour day is broken down into 2 hour segments that line up approximately correct most of the time. Also of note, is the year, which is the old Imperial year which started at 660 AD so it is 660 years ahead of western calendars.

Experience in implementation of design patterns:

My experience was hit or miss. The first night of work I spent 4 hours drawing out the UML to ultimately scrap the specifics of the classes because I did not know as much as I needed about android to accurately plan. The interactions proved useful and kept things in order when things could have gotten out of hand very quickly due to Java in Android Studio demanding more and more shenanigans to get simple things to work. Such as trying to update adaptors and all the code needed to implement simple dropdowns was immense. Building the actual designs were fairly easy to implement once Java for Android is taken care of. I can definitely see their use on massive projects to keep everything in order and not have any devs suddenly change one class’s functionality just because it would be easier for them to do so.



Hour 2
Minute 19
Second 32
Weekday Thu..
Day 11
Month Sep
Year 2018

Thursday, 11, September, 2018. 2:19:40

mokuyōbi, jūichinichi, Nagatsuki, 2678. Hour of the Ox:19 pun, 40 byou.

mokuyōbi, jūichinichi, Nagatsuki, 2678. Hour of the Ox:19 pun, 40 byou.

Thursday, 11, September, 2018. 2:19:40