1D Java Report

Group: 2-8

Gui An 1002651 Kenneth Soon 1002691
Jun De 1002682 Tan Jin Yan 1002722
Yi Dan 1002954 Sumedha Gn 1002876

Background and Problem:

3D printing jobs continue to have a high rate of failure. Due to its high uncertainty of success, this incur unnecessary wastage of resources and time. 3D Printer Mate recognizes and attempts to minimize this uncertainty to improve printing experience amongst students.

System Design and Implementation:

Our project focuses more on the modular implementation. We made our monitoring system separate from the 3D printer rather than integrating them. This is a practical approach because it makes our project adaptable across any 3D printer.
For the Firebase, to ensure that the data transferred and updated on it and to the app takes the fastest time possible, we minimized the data stored in Firebase in consideration of mass application. In the context of implementing large numbers of monitors, this minimizes the overhead of our project.
To minimize data usage and preventing database to be overloaded, we only kept the latest status of the printer and the printer's sensor raw values. The image of the printing process is also refreshed every 15s and the latest image will overwrite the current image in Firebase.

Android App

All constants and global variables are kept in files separated from all android app activities. String constants are kept in 'R.string' while global variables are kept in 'SharedPreferences'.

We created Listeners that constantly listen to changes made in Firebase. Once at the beginning to retrieve the index number of all existing 3D printers (for user to log in), a constant listener at the background to send notifications when the print has stopped, and other constant listeners to update the status of the 3D printer in the app, and update the image of the 3D print in the app.

Our overall design of the Android application places the user's focus not on the android application, but on the notification it provides. This is a deliberate decision as it will be inconvenient to open the app just to check the status of the 3D printer. This is proven during our exhibition when many visitors agreed that focusing on notifications makes this project an attractive product to adopt.

Hardware

To incorporate the hardware together, we used a mixed of Java, Python and Terminal implementations. The hardware used are as shown below:

Sensor suite:

1. A SRF05 Ultrasonic Sensor, accurate 4mm-3000mm+-1mm
2. A SR04 Ultrasonic Sensor, accurate 4mm-3000mm+-1mm
3. An RPM Sensor, IR-based, counts number of times IR beam breaks between its two terminals, has a volatile byte
4. Arduino Nano, ATMega 328
5. Raspberry Pi 3 B+ (WiFi-enabled)

Hardware implementations:

1. Java:
    a. Custom class implementation of JSON due to missing libraries (JSONObject.java)
    b. Data compilation done with File I/O, integrated terminal script, and ImageIO objects
    c. DatabaseProcessor.java handles calling of individual modules and the data needed, wraps it into a JSON object, opens HTTP connection to post to firebase via RESTful API.
2. Python:
    a. Reads serial port output from Arduino sensor via pySerial library
3. Terminal:
    a. Installed fswebcam application package - handles streaming of webcam data and saving into an image. Can be timestamped, but for the sake of visuals it’s left out.

Possible future implementations

During our initial discussions, we have thought of integrating our Rpi to the arduino chip inside the 3D printer to remotely control it, only when we have the necessary implementations done. From here, we can expand our work into more printers in school. This can be achieved by assigning each printer its own printer ID and creating an admin account that may monitor status of the 3D printers. On the Firebase, we can integrate more 3D printers (and their sensors) accordingly.

To further improve our current idea, it is possible to use structured light and OpenCV to map out a 3D shape of the print, compare it with the file uploaded into the Arduino in the chip and notify the user if there is a huge difference between the 2 models. If so, then the app will notify the user that the print has failed.

Lessons learnt

The backend of our project was initially very ambitious. Our original plans was to have a PostgresSQL database to host the data posted by the Rpi. This allows our project to be even more modular, not relying on specific Google services but any database that runs on PostgresSQL to host our project. We encountered many challenges trying to do so as we weren't sufficiently adapt at using AWS or GCP cloud servers that host these databases. Given such a tight dateline, we would not be able to make much progress so we switched to Firebase. We should have either used Firebase from the beginning, or hosted a PostgresSQL database on one of our computers instead to maximise our time.

Also, we were inexperienced in segmenting workload which prevented us from working in parallel. Our experience in this project allows us to better identify ways of segmenting project workloads to work more efficiently in the future.

Conclusion

Overall, this project has gave us the chance to practice on the functions both taught in class and offered beyond the class. The possible ways to implement IoT was also thoroughly explored during our ideation phase and hence, this project has been a very fruitful one to all of us.

References:

"Home." *PostgreSQL Tutorial*
www.postgresqltutorial.com/

"Structure Your Database | Firebase Realtime Database | Firebase." *Google*, Google
firebase.google.com/docs/database/ios/structure-data

tutorialspoint.com. "Firebase Google Authentication." *Www.tutorialspoint.com*, Tutorials Point
www.tutorialspoint.com/firebase/firebase_google_authentication.htm

Rashmin, Ravindu. "Arduino to Android , Real Time Communication For IoT with Firebase." *Medium.com*, Medium, 22 July 2018, medium.com/coinmonks/arduino-to-android-real-time-communication-for-iot-with-firebase-60df579f962

Instructables. "Firebase Integrate With ESP8266." *Instructables.com*, Instructables, 27 Dec. 2017, www.instructables.com/id/Firebase-Integrate-With-ESP8266/
"Connect to Firebase | Android Developers." *Android Developers*, developer.android.com/studio/write/firebase

New to Arduino - Is a Computer Required ??
forum.arduino.cc/index.php?topic=288522.0