Coding Tasks

1. Lightweight LSTM

Target platform: Android
Software: Android Studio
Language: Java/Kotlin, Python

Description:
In this task, you are asked to develop an Android app named “Task 1” such that, once a button is pressed, it reads a time-series datafile, encodes the data into a compressed embedding using LSTM, and saves the output as a file named “output.xxx” (xxx = csv, txt, xls, etc.) in the same folder.
Download the original datafile (‘EyeQoE_Dataset/0/010.csv’) from:
https://github.com/MobiSec-CSE-UTA/EyeQoE_Dataset/blob/main/0/010.csv
In this datafile, each row indicates a timestamp and each column specifies a head/eye movement behavior. You are asked to implement a light-weight LSTM model to learn from the gaze movements exclusively, which means **the input will be the 5th and the 6th columns** in the data file, i.e., “gaze_x” and “gaze_y” (input size: t * 2, t = # timestamps). The output should be a **vector of length 50**.
Use **TensorFlow Lite** framework for the LSTM model. You can initialize it using Python and store the model as “model.tflite”. You don’t need to train the LSTM model; you can simply initiate it with random variables. **The model’s performance is not important.**
Make sure your app is **error-free** and can be run on an Android smartphone.

Deliverables:

- Package of the Android app “Task 1”
- The output file “output.xxx”
- A README.txt file with necessary instructions

2. Data Uploading

Target platform: Android
Software: Android Studio
Language: Java/Kotlin

Description:
In this task, you will need to develop an Android app named “Task 2” such that, once a button is pressed, the target file is uploaded into Google Drive. If you successfully generated the output file “output.xxx” in Task 1, upload this file. Otherwise, upload the original file “010.csv” (use the same link above to download). You’ll need to manage with Google’s API key. Make sure your app is **error-free** and can be run on an Android smartphone.

Deliverables:

- Package of the Android app “Task 2”
- A README.txt file with necessary instructions

3. Acceleration Data Display

Target platform: Android
Software: Unity
Language: C#

Description:
In this task, you are asked to build an app named “Task 3”, which continuously measures the current acceleration in three axes of the smartphone from its IMU sensor and displays it on the screen. You need to use Unity and create a 3D project, and build the project for the Android platform. Make sure your app is **error-free** and can be run on an Android smartphone.

Deliverables:

- Package of the Android app “Task 3”
- Package of the Unity project
- A README.txt file with necessary instructions

Submission guidance

- Submission deadline: **June 25th by 11:59 PM**
- Submit all your documents as a **zip file** named by your last name, e.g., “Zhu.zip”
- You can use my email address below to send me either the zip file directly (if size allows), or a download link (GitHub, Google Drive, OneDrive, etc.) to your zip file
- **Collaboration is strictly prohibited**

If you have any questions, don’t hesitate to contact me at huadi.zhu@mavs.uta.edu.