Android documentation

Data flow

1. In Server.java line 78 in the while loop, data [yaw, row, pitch, acceleration x, acceleration y, acceleration z, indicator] is received over socket.
2. In Server.java line 91 & 92, it sets the gyroData (float[3], stores yaw, row, pitch) and indicator (boolean, 1=button_released, 0=button_pressed) variables in GestureTypingService.java.
3. In Server.java line 95, map the row & pitch angles to the coordinates on the android keyboard.
4. In Server.java line 96, call ChatHeadService.moveChatHead() to move the ball on UI
5. When setIndicator() is called from step 2, in GestureTypingService.java line 122-141, detect when the button is pressed, and whether it's a short click or long press. Store them in the buttonPressed, buttonShortPressed and buttonLongPressed flag.
6. In GestureTypingService.java line 181, the while loop constantly runs.
7. In GestureTypingService.java line 182, convert the xy coordinate values in chatHead.java to the coordinates of shell commands (will need this coordinate later when we call sendevents(), which calls the function jni/sendevent.c)
8. In GestureTypingService.java line 229, if button is pressed and the current state is typing, then it means user has finished typing a word. So we send finish typing events and close the file.
9. In GestureTypingService.java line 239, if the button is short pressed and the current state is idling, there are three cases: i) user click to select a word. ii) user click the next key (the delete key in TEMA keyboard) to go to the next phrase. iii) user starts typing. Which of the three operations is performed is determined by the coordinates of the cursor in the keyboard.
10. In GestureTypingService.java line 261, if the button is long pressed, it means user want to delete the last word. Fetch the last word length by calling getLastWordLength(), then call doDeleteWord().
11. In GestureTypingService.java line 271, if the current state is typing, we use timestamps to control calling sendevents(). We send the new coordinates to sendevents() every 50 ms. Here setting a time gap that's too small or too large will result in the gueture typing trace being not smooth.

Coordinates convertion

There are three set of coordinates:

1. The gyro data that represents angles (only row and pitch are used). Imagine the angles in row represents x and the angles in pitch represents y.
    - Boundaries are defined in MainActivity.java ARDUINO_RECT variable
2. The x and y coordinates in ChatHeahService.java. The coordinates is used for diplaying the ball (cursor) in UI
    - Boundaries are defined in MainActivity.java CHATHEAD_SELECTWORD_RECT, and GestureTypingService.java CHATHEAD_RECT and CHATHEAD_SELECTWORD_RECT variables. Note that the CHATHEAD_SELECTWORD_RECT variable includes the select word section in the keyboard.

3. The x and y coordinates in GestureTypingService.java (example: line 182 newCoords variable). These coordinates are used when we send shell commands to functions in sendevent.c. You can find these coordinates by i) Go to Settings > developer options > turn on 'show touches'. ii) multiply the x, y values shown on screen by 2.
   - Boundaries defined in GestureTypingService.java KEYBOARD_RECT and KEYBOARD_SELECTWORD_RECT variables. Note that the KEYBOARD_SELECTWORD_RECT variable includes the select word section in the keyboard.

Util.rectToRectMapping takes two rectangles (rect1 and rect2, specified by the boundaries, which are min_x, max_x, min_y, and max_y), and the x and y of one point in rect1, and returns the mapped x and y values in rect2.

Notes:

1. In GestureTypingService.java, you can ignore anything inside if (continuousPressTyping) {} becuase we're not using the continuous typing method.