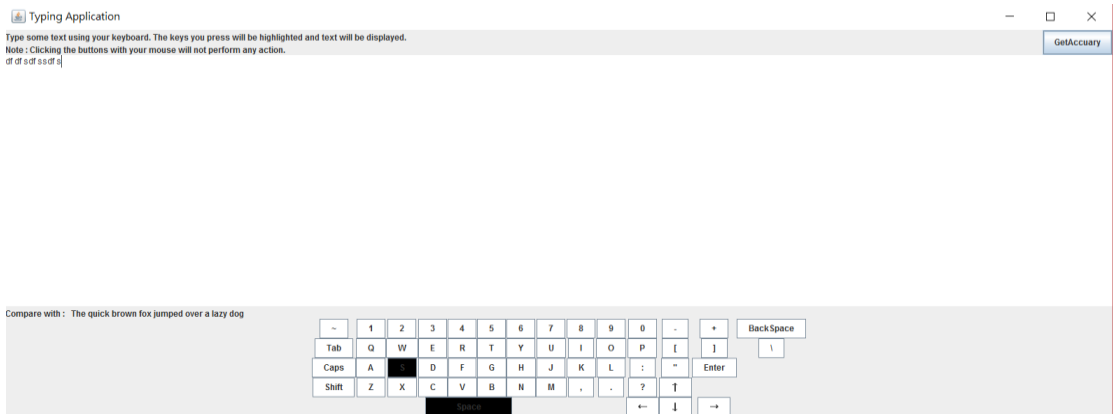


1. 該 `typing tutor class` 是繼承自 `jframe`，並且 `implement keylistener` 介面用來監測使用者輸入的鍵盤動作
2. 程式有 2 個主要 `private variable` "`Sample_text`" 和 "`keyCodeButtonMap`"，前者儲存對比的字串，後者儲存鍵盤上的鍵值分別對應哪個鍵盤的按鍵
3. 在 `constructor` 裡面，一開始設定該 `frame` 的背景色為白色，並且宣告所有的鍵盤上從第一排到最後一排的按鍵；並且用 `JPanel` 把這個 `frame` 分成北、中、南三部分，北區域放置說明訊息和"`getaccuracy`"按鈕，中區域放置使用者輸入的訊息，南區域放置虛擬鍵盤；北、中兩區域都有使用 `method setLayout` 和 `method BorderLayout` 來做排版的動作，而南區域的虛擬鍵盤則是用 `GridBagConstraints method` 將按鍵排版，分在不同列和調整間距等等，而每次將按鈕的值放入 `Jbutton` 後，會把用於 `keylistener` 中的 `KeyEvent` 的對應數值分別是哪個鍵盤的按鍵用 `keyCodeButtonMap` 存起來，(鍵值的部分則是使用自定義的 `getKey method` 取得)，並將該 `button` 背景色設為白色；而在 `getaccuracy` 按鈕動作部分，把字串比對完之後，將計算結果使用另一個新開啟的 `Jframe` 來顯示
4. 在實作鍵盤按下的動作部分，使用 `keyevent` 的 `getKeyCode method` 來取得使用者按下的鍵值，再用 `keyCodeButtonMap` 取的該鍵值是對應到哪個 `Jbutton`，再將此 `button` 使用 `setBackground method` 將該 `button` 顏色設為黑色
5. 實作鍵盤釋放的部分，也是使用 `keyCodeButtonMap` 取得對應鍵盤按鍵，再將該按鍵的顏色設為一開始設定的白色，取得該按鍵原本的背景色使用 `getBackground method`
6. 實作 `keyTyped` 的部分因為沒有要幹嘛所以讓它空著

7. 鍵盤按下後反黑虛擬鍵盤，釋放後還原背景色



8. 計算正確率

