Gathering data

In order to have relevant and useful conclusions to the program, we had to figure out the means to produce precise data. We used python's ‘tkinter’ library combined with a widget to record keyboard events. With the help of the library we could easily record the time it took to press a certain key, but also measure the time between each keypress. We did this by recording the current time in nanoseconds each time an event happened and calculated the relevant differences.
For example, in order to get how long a key was pressed, we record the time it was pressed and the time it was released, and calculate the time difference between the two.

```
time_to_press.append((presses[i][0],releases[j][1] - presses[i][1]))
```

For the time between keypresses we measure the release of the first key and the press of the second key, and take their difference.

```
times_between_keys.append((presses[i][0],presses[j][1] - releases[i-1][1]))
```

We decided to gather the data in two lists of tuples (one for keypress time and one for time between key), where the first items are the letters and the second items are time measurements in nanoseconds.
Initially we ran into a few problems. When the user types fast enough, the inputs can get mixed up, because the following can occur: the user can press a key earlier then the previous key was released, thus resulting in a mixed order of input. This way the wrong letter would get the corresponding time measurement which leads to faulty data.
To circumvent this we had to use a nested loop which loops over the presses and the releases and only records the data if the ’i’-th pressed letter is the ’j’-th released letter (pairs up the releases to the presses):

```
for i in range(1,len(presses)):
        j = i
        found = False
        while j < len(releases) and not found:
            if presses[i][0] == releases[j][0]:
                times_between_keys.append((presses[i][0],presses[j][1] -
releases[i-1][1]))
                found = True
            j = j + 1
```

This way we could easily record and push the data into an excel table where each of us had 10 attempts to write the words ‘apple’ and ‘university’. Each row has a designated user and letter, and an attempt consists of two columns (keypress and time between keys).