Capitalization Convention Experiment

Experiment 2, Experimentation & Evaluation 2021

Abstract

In this experiment we tried to find out whether camel case or kebab case is more readable. The test consisted of reading two or more words that appeared on the screen and choosing the correct answer as quickly as possible.
Two questions on the first page had to be answered before the test. The questions were about the input device and whether you work or study in computer science.
We used three independent variables: word case (camel or kebab), word sequence, and correct response index.
During the experiment, we also measured the time it took the person to respond, starting from when the start button was pressed after reading the word, and whether the response was correct or not.

1. Introduction

The experiment was conducted with the aim of finding whether it is faster to read words in camel case or kebab case.
Since programmers read many words one wants to try to find out which adopted case is more efficient in reading.
This could improve programming efficiency and code readability.

Our proposed experiment will show 2 or 3 words separated by space on the screen, and only after pressing the start button will the 4 buttons appear where there will be the same words but separated using kebab-case or camelCase.
Between the click of the start button and the one that is considered correct will be calculated the time,to analyse the data later.

Hypotheses:
Hypothesis 1: People familiar with computer science and programming will perform better at reading words formatted in camelcase and kebabcase than people without this experience.
Hypothesis 2: It is important to have a well defined ‘space’ character when reading since it makes the difference between words more noticeable at first glance. Therefore, we hypothesise that kebabcase will perform better than camelcase in general.

2. Method

2.1 Variables

Independent variable	Levels
Case	(Or capitalization convention) Camel case (CC), Kebab case (KC) To facilitate design, participants are part of one of two groups (or sets): participants of the first group receive tasks with a predefined case, while the other group receives tasks with the other case. For example, Group 1 receives tasks in the following order: CC KB CC CC KB… Group 2 receives tasks in the following order: KC CC KC KC CC… Each group contains the same number of KC and CC instances. In the data analysis part, we don't consider the group but rather the case associated with each data point.
Correct Index	The index from 0 to 3 of the button in which the correct answer is located. To facilitate design, participants are part of one of two groups: each group (or set) has a predetermined order of the correct index for every test instance. Each group contains the same number of 0, 1, 2 and 3 correct indices. 0: Top left 1: Top right 2: Bottom left 3: Bottom right In the data analysis part, we don't consider the group but rather the correct index associated with each data point.

Dependent variable	Measurement Scale
Time for correct click	The time (in milliseconds) taken to click on an answer for each question
Correctness	Whether the participant clicked on the button with the correct index for each question

Control variable	Fixed Value
Word sequences	The words that appear on the screen and the correct spelling. Each participant answers each of the 24 questions in the same order as every other participant. For the complete list of word sequences used, see the section on design.

Each participant is asked to answer two questions that we felt were pertaining to the experiment:

Blocking variable	Levels
Computer science experience	Whether the participant has studied or works in the computer science field. (Possibilities: Yes or No)
Input device used	Which device the participant is going to use for the experiment. (Possibilities: Trackpad, Mouse, Touchscreen)

2.2 Design

Type of Study (check one):

□ **Observational Study**	□ **Quasi-Experiment**	x **Experiment**

Number of Factors (check one):

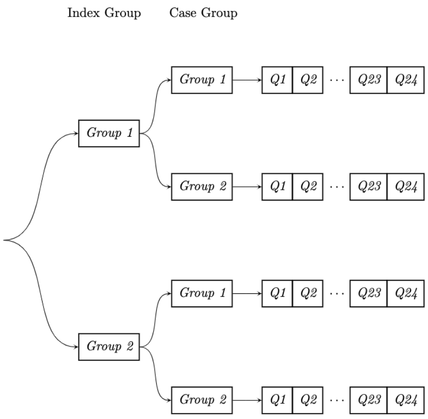
□ **Single-Factor Design**	x **Multi-Factor Design**	□ Other

Between vs. Within (check one): [for human subject studies]

x **Between Group Design** (independent measures)	□ **Within Subject Design** (repeated measures)	□ Other

We designed the experiment as a web app that each participant would access to in order to conduct the experiment. For each question the participant answered, the frontend would gather data from the participants and send back the results to the backend, and would then request the next question of the experiment. The frontend and backend would therefore communicate via a REST API.

Below we can see a diagram of how the participants are divided in groups:



This means that the experiment follows the “Full factorial design” principle, where every possible combination of independent variables is tested. At the end, we have $2 \times 2 = 4$ total possibilities. With 24 participants we are able to perform 6 tests for each combination.

Each participant is then asked 24 questions one after the other. The word strings can be found in the web_app/public/words.csv file, where each row represents a question.
A row is formatted in the following way:
is1, is2,defcase,correct_spelling,option_1,option_2,option_3
Where:

- Is1 is the correct index if you are part of Group 1 of the correct index variable
- Is2 is the correct index if you are part of Group 2 of the correct index variable
- Defcase is the capitalization convention presented to you if you are part of Group 1 of the case variable (If you are part of Group 2 you are presented the opposite convention)
- Correct_spelling is the sequence of words you are trying to find
- Option_1 is the first incorrect option
- Option_2 is the second incorrect option
- Option_3 is the third incorrect option

The correct spelling of the words used for the experiment are the following:

event bus
add event listener
get by class
cookie clicker
word array
is outer face
ruby is cool
orientation test
move over there
conjugate gradient descent
medial axis transform
large data set

feedforward neural network,
counter clockwise
origin point
euclidean distance
turing machine,
singleton factory
train test split
up down right
delaunay triangulation
sin cos tan
new fixed thread pool
terminate experiment!

2.3 Participants

24 Participants (assigned UID 0-23) took part in the experiment, and their identity remained anonymous.

The participants were recruited from various places: some are classmates, others are friends and some are family members or colleagues of family members.

Very little information was collected about their identity, only what is shown in the Blocking Variables section. In particular their computer science experience, which we felt may have an impact on their performance (since those who are not used to programming may have a hard time parsing these capitalization conventions).

The participants were assigned to the groups described in the Independent Variables section randomly, but making sure that there were about the same number of participants in each possible combination of groups:

In total there were:

- 6 People in Case Set 1 and Index Set 1
- 6 People in Case Set 1 and Index Set 2
- 6 People in Case Set 2 and Index Set 1
- 6 People in Case Set 2 and Index Set 2

2.4 Apparatus and Materials

Each participant was able to use their device and input method of choice when performing the experiment.

The web application used can be viewed and downloaded from our GitHub page. It was designed such that it's easy to add more questions to the words.csv file and have them automatically appear.

The javascript method `window.performance.now()` is used to measure the time it takes a participant to click on an answer.

Before the data is sent to the backend, it is stored in the LocalStorage of the browser. This means that it's possible to perform the experiment again by clearing the LocalStorage or by accessing the page from another browser.

In case anything goes wrong with the upload of the data to the backend, the route '/resend' can be used to send the data again.

2.5 Procedure

Each participant received a link to the website which hosted the experiment. They each had to read the instructions, complete the form which asked the two blocking variables, and then they were able to try 3 instances of the experiment before beginning.

Behind the scenes, the backend would assign each participant one of two "index sets" and one of two "case sest". These are the groups that are explained in the "Case" and "Correct Index" segments of the Independent Variables section.

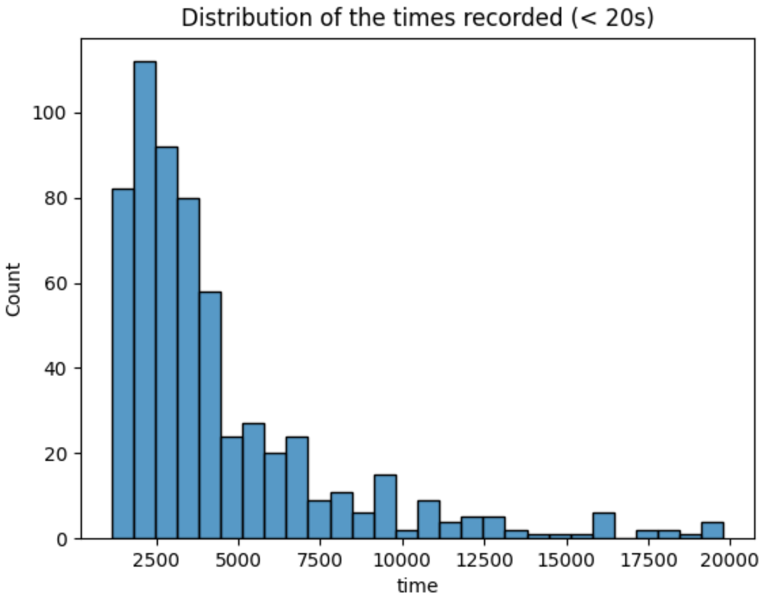
Each participant was then given a Word Sequences and asked to identify the correct spelling over 4 possibilities. The correct spelling was given first, and then once they pressed a button to continue the counter would start and the choices would appear. The choices were laid out in a 2x2 grid.

Once a participant clicked on a spelling, their answer and time were recorded and the next Word Sequence would appear until there were no more. In total, each participant performed 24 of these tests.

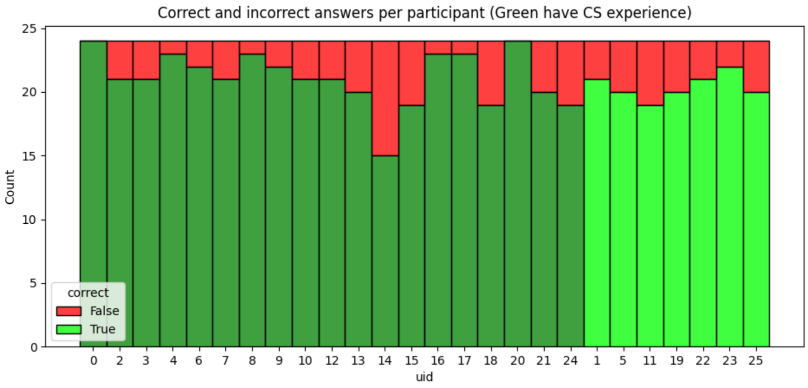
3. Results

3.1 Visual Overview

(Note that some of these graphs may be slightly different from the final results in the final “/figures” folder of the repository)

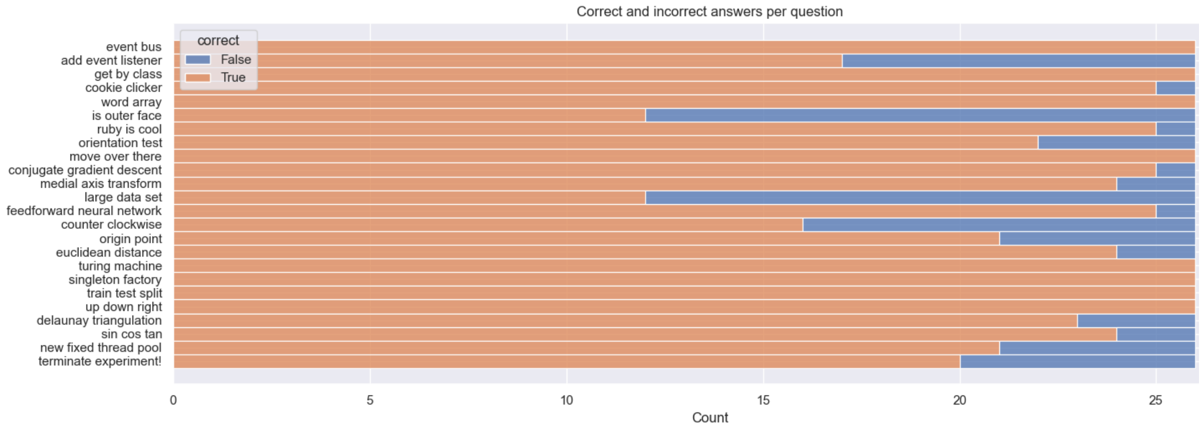


(1) The above figure shows the overall distribution of the times under 20 seconds (All time values shown in this section are in milliseconds), both correct and incorrect. Most of the recorded times are around 2 to 3 seconds.

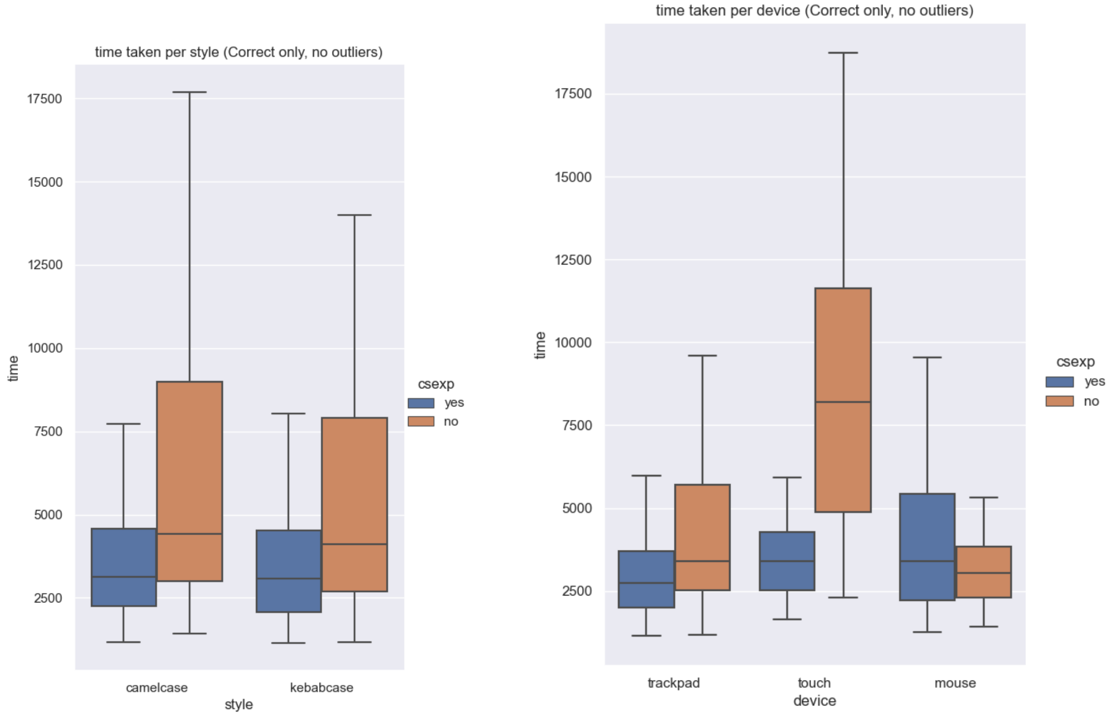


(2) The above figure shows the number of correct and incorrect guesses for each participant.

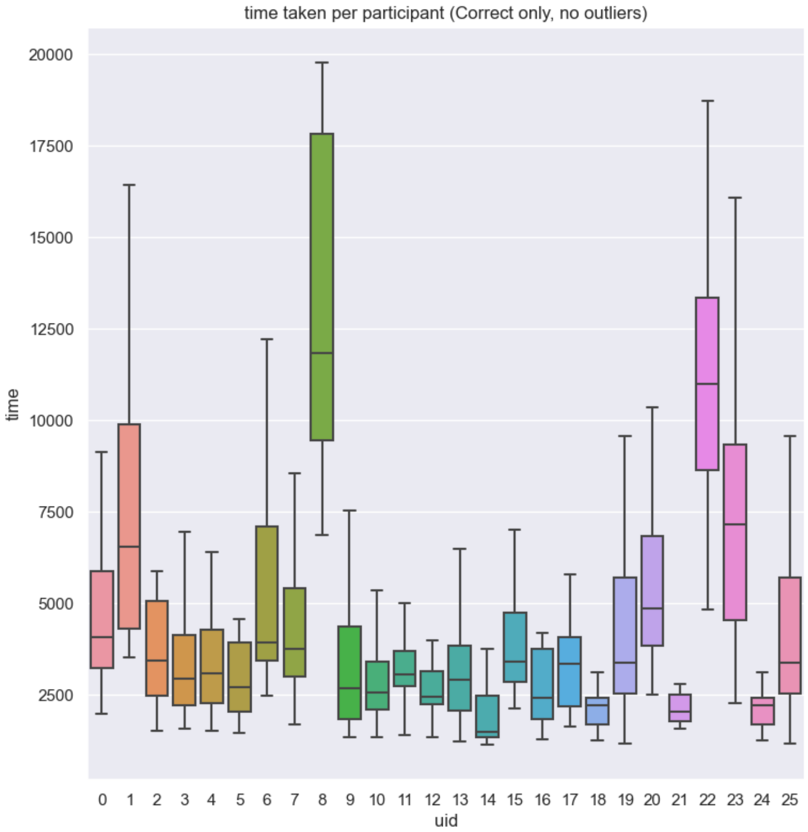
The dark green bars are the participants who have studied or work with computer science, while the lime bars do not.



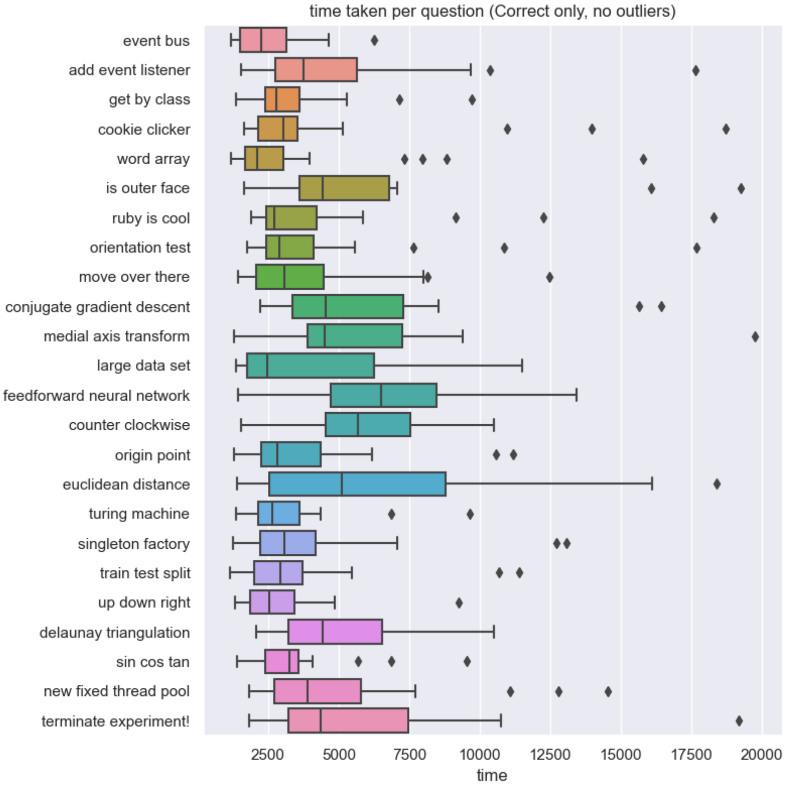
(3) The above figure shows the number of correct and incorrect answers per question.



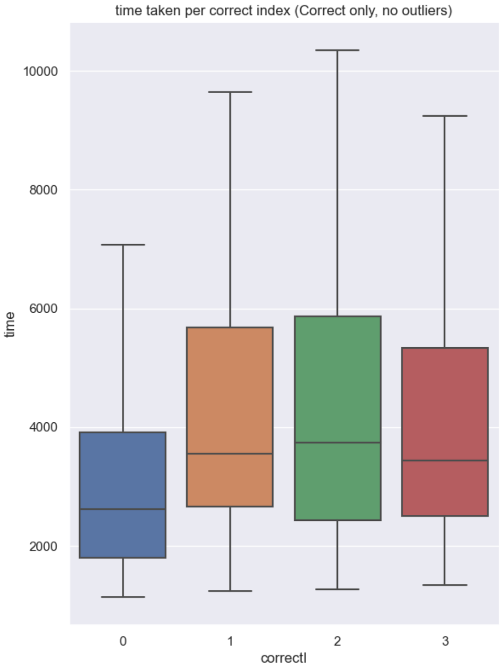
(4) The above figures show the box plot of the time taken for participants to answer questions for each convention (left) and depending on their input device (right). The orange boxes are times from people with computer science experience and the blue one are times from people without computer science experience. These times exclude the clear outliers (above 20 seconds) and the incorrect answers.



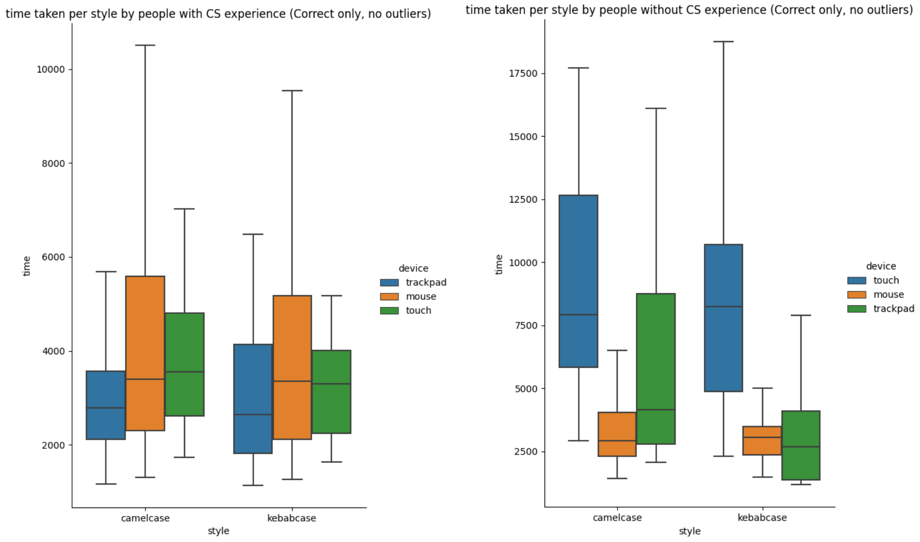
(5) The above figure shows the boxplots of the times from every participant. Times above 20 seconds and incorrect answers are excluded.



(6) The above figure shows the boxplots of the times from every question. Times above 20 seconds and incorrect answers are excluded.



(7) The above plot shows the times recorded for each correct index. Recall that a correct index of 0 means that the correct solution was the one in the top-left.
Notice how having the first index be the correct one causes times to be faster, likely because most people read left to right and would recognize the correct answer before reading the rest.



(8) The above figures represent the performance of people with (left) some computer science experience and (right) no computer science experience.

3.2 Descriptive Statistics

The provided Python script, apart from generating the images used in the Visual Overview section, also calculates the “five-number summary” for each of the experiment instances of the Total Time dependent variable. Below is a table of some of the instances together with their summary.
Shown is the data for devices mouse and trackpad, and considering any correct index. The file also generates data for the first index being correct only and for all but the first index being correct.

Device	Case	CS experience	Min	2nd Quartile	Median	3rd Quartile	Max	Mean	STD
mouse	kebabcase	yes	1260.0	2142.0	3437.0	5764.5	43784.0	5270.559	5939.996
mouse	kebabcase	no	1463.0	2342.0	3052.0	3488.0	7538.0	3254.85	1394.398
mouse	camelcase	yes	1306.0	2342.0	3581.0	6300.75	37584.0	5588.906	6252.794
mouse	camelcase	no	1416.0	2309.5	2904.0	4048.5	6839.0	3308.632	1444.028
trackpad	kebabcase	yes	1134.0	1823.75	2639.5	4133.5	10680.0	3399.934	2148.675
trackpad	kebabcase	no	1166.0	1375.0	12685.0	4102.0	7896.0	3191.222	2000.366
trackpad	camelcase	yes	1164.0	2111.75	2783.5	3571.5	11521.0	3134.273	1774.194
trackpad	camelcase	no	2070.0	2777.25	4144.0	8738.75	16105.0	6152.818	4394.577

The rest of this data can be viewed by running the Python script provided in the Github repository.

3.3 Inferential Statistics

Just by looking at our graphs, it’s not clear if our hypotheses are correct.
Therefore, we will perform a t-test to check if we can reject the null hypothesis for of our own. If we can, then we confirm with the mean that the better results are given by the set that we expect.

First hypothesis: People familiar with computer science and programming will perform better than those without.
First we compute the mean, the standard deviation and the square root of the size of the data from people with computer science experience:
μ = 4574.587268521617
σ = 4955.016437731484
$\sqrt{N}$ = 20.784609690826528
Then we compute the mean of the data from people without computer science experience:
$\overline{X}$ = 7378.347916654415

And compute the z-score like this:

$\frac{\bar{X}-\mu}{\sigma/\sqrt{N}} = 11.760822889342144$

Since the z-score is large enough, we can reject the null hypothesis and take the set with the lower mean as the one that performs better.

4. Discussion

4.1 Compare Hypothesis to Results

In section 3.3, we saw that our first hypothesis can be confirmed statistically

However, our first hypothesis can be rejected by simply looking at the graphs. We can see in Figure 4 (left) that there is barely a difference between the two box plots, meaning that the overall performance is similar. Even for when we account for computer science experience and input style, Figure 8 shows that the only discrepancy in favour of kebabcase (with input type trackpads) could just be explained by people with CS experience having more practice with trackpads.

4.2 Limitations and Threats to Validity

Some of the queries were difficult on purpose, however some were maybe too difficult. For example the ones where none of the options were misspelt but rather the capitalization was incorrect (e.g. isOuterFace vs IsOuterFace): this element was not explained correctly and the instructions only mentioned that the options were misspelt.

One possible way to use this in the experiment's favour would have been to categorise users based on how quickly they respond to these difficult questions: If they answer quickly it means that they click on the first option they read which *looks* correct, if they answer slowly it means that they read all options multiple times and may be confused by the question.

4.3 Conclusions

In conclusion, it's clear that a sample size of 24 participants is not large enough to be able to gather conclusive evidence, with the only thing being proven for certain is that people with experience in computer science are able to recognize both camelcase and kebabcase faster when compared to people without experience in computer science.

Appendix

A. Materials

A short informed consent document is present in the website where the experiment was hosted, as well as all of the instructions required to perform the experiment.

The following resources helped in the making of this report:

- The information document about the assignment.
- The seaborn and pandas python libraries
- Express, a Node.js web framework
- Microsoft Azure, for hosting the web app

B. Reproduction Package (or: Raw Data)

The raw data can be found in the GitHub repository in the web_app/public/final_data.csv file. To generate the plots seen in this report, follow the instructions in the README file to run the python script, or unzip the archive provided (figures.zip).