SCRATCH TANGIBLE INTERFACE

Human computer interaction unit 2017
University of Bristol

Callum Fawcett
cf16670@my.bristol.ac.uk
Computer Science

Andreas Hadjiantonis
ah16036@my.bristol.ac.uk
Computer Science

Liam Hunt
lh16454@my.bristol.ac.uk
Maths and Computer Science

Kenny Lomas
Kl16942@my.bristol.ac.uk
Computer Science

Faizaan Sakib
ss16161@my.bristol.ac.uk
Computer Science

Felix Williams
fw16084@my.bristol.ac.uk
Computer Science

Angus Williams
aw16797@my.bristol.ac.uk
Computer Science with Innovation

Short summary
The main idea of the quantitative experiment that we decided to pursue was to see how the user would perform a task in a certain environment on both a digital interface and a tangible interface. With this decided, we had to find a task that we would be able to translate between the digital and the real world so we could get meaningful data out of it.

We were inspired by existing systems such as the Cubetto[1], introducing programming to children in a user-friendly way. Also, Instructiblocks[2] gave us an insight to how tangible and digital interfaces can interact to allow the user to have a more enjoyable and creative experience. This led us towards a wish to explore the possibilities of how interaction between different interfaces can aid children at doing particular tasks.

In her paper entitled" Tangible User Interface for Children An Overview"[3], Diana Xu talks about how it requires little cognitive effort to learn how to use TUIs(tangible user interfaces), and therefore less time, and how TUIs support "Trial-and-Error" activities. This idea of using TUIs as a superior learning tool is something we were interested in, but ultimately did not approach in this experiment. Xu also speaks about how TUIs can support multiple users at a time, an interesting factor we hadn't thought of.

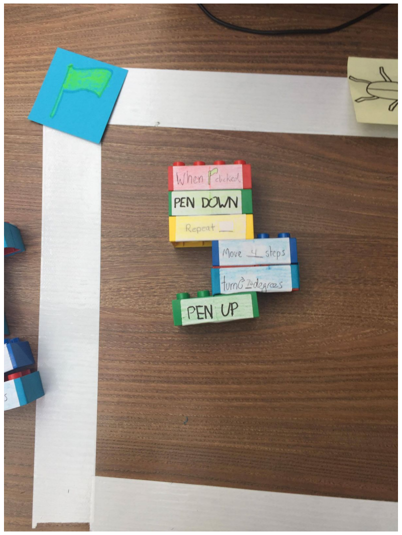

Fig.1 Photo of tangible interface

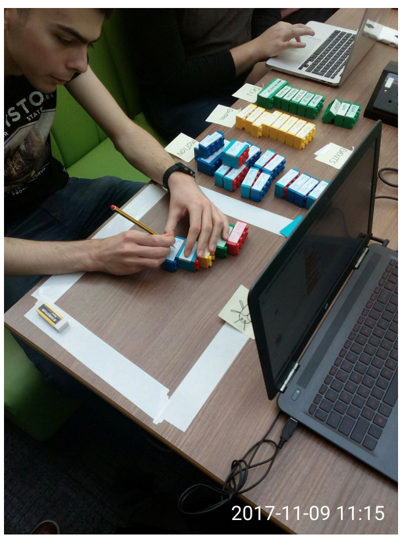

Fig.2 Photo of tangible interface

In the end, we decided that our experiment would see how users would fare with using the programming language Scratch, on the ordinary digital interface and a tangible interface of our own creation. Scratch is a visual programming language, that uses modular code 'blocks' that fit with each other that can then be executed to be performed by a 'sprite' in a 2D environment. With its intuitive and child-friendly functionality, along with its modular nature, we thought would be perfect to test our idea on. This experiment will help us to see which method is quicker to solve problems and whether or not the 'interactive' element of the tangible interface would change much compared to the existing interface.

In order to create a tangible interface for Scratch, we used coloured building blocks to replicate the code blocks used in the language, with each colour identifying their category just like in Scratch. We created this as a 'Wizard of Oz' prototype, where the actions on the tangible interface would be simulated on the digital interface in real time.

The results we obtained were not sufficient to disprove our null hypothesis, but we did find some interesting insights in our results.

Hypothesis

To be able to quantify the performance and effectiveness of either interface, we needed a dependent variable that is unambiguous and well-defined. Several aspects of the user's performance can show this but most of these are hard to quantify, and the results may be introduced to bias as they may very much depend on the user's individual programming skill. Furthermore, the open-ended nature of these variables would require more time to be given to each user, which was not abundant.

Relative to the other options, time is a much more quantifiable measure of performance. We decided of the following Null and Alternative hypotheses:

$\mathbf{H_0}$: *"Any observed differences in time taken to complete a task in Scratch using the digital interface and the tangible interface are due to chance alone."*

$\mathbf{H_a}$: *"It takes less time to complete a task in Scratch using the digital interface than using the tangible interface."*

Although the tangible interface may be more engaging or even more natural (e.g. putting actual blocks together rather than dragging blocks), we believed that users on the digital interface would require less time to complete a task than users on the tangible interface. This was due to several factors. Firstly, although it may be more subconscious, it may take some time for users to acclimatise to a new realm of 'tangible' programming, something which most people may have not done before. Also, although moving and putting blocks together may feel natural, using a mouse will allow the user to navigate the environment much quicker. A fraction of a hand movement, can equate to a large movement of the arm from screen to real life.

Similarly, the user may have to 'insert' variables or values into their code. In this case, typing and being able to instantly clear/replace these values would take less time than using a pencil to write them down, or have to erase an old value with a rubber before proceeding.

Finally, since Scratch was designed by experienced programmers with specific goals in mind with extended time and testing, compared to our prototype, we would expect the default digital interface to yield better results.

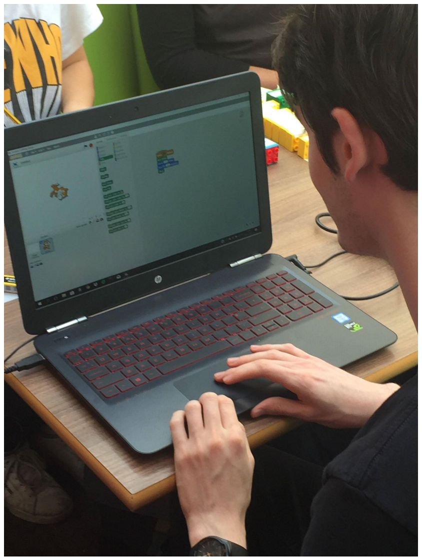

Fig.3 Photo of digital interface

Experimental design

Participants

For our experiment, we used 30 participants to gather enough data to be able to make conclusions and test the hypothesis thoroughly. These participants were split into two evenly sized groups of 15. One group would be tested on the digital interface, the other on the tangible interface. Each group was given exactly the same task and only tried the experiment once as trying it multiple times would result in them knowing what to do and therefore being able to complete the task almost instantaneously. Each participant was questioned on their previous experience with Scratch so that we could separate if necessary the people who had a lot of experience from those for whom it was their first experience of Scratch.

Variables

The Independent variable for this experiment was the interface used: the tangible interface and the digital interface.
The Dependent variables were:

- The time taken to perform a given task (in seconds)
- The number of failed program executions (when the green flag is pressed but the program is incomplete/incorrect).
- The participant's Scratch Proficiency (on a scale 0-3):
  - 0:Never used Scratch.
  - 1:Has used scratch, but not in the last 24 months.
  - 2:Has used Scratch, within the last 24 to 12 months
  - 3:Has used Scratch in the last 12 months.

To improve the reliability of our experiment we maintained the following Control Variables:

- Same task for each interface (draw a pentagon).
- Experiments conducted in same environment.
- Same introduction to Scratch before task starts.
- Same start/pause/end points for timing across both interfaces.

Materials

In conducting the experiment, we used 2 computers (with Scratch program installed) for digital interface testing (see fig.3) and for the Wizard in the tangible interface experiment respectively.
We also used labelled LEGO blocks grouped by colour to represent programming blocks in Scratch (see fig.1 and fig.2).

Task

The participant must draw a pentagon (ie a five sided polygon, each side being the same length) anywhere on the screen. The pentagon can be of any size. The participant may only make use of the blocks in the categories specified: event blocks, motion blocks, pen blocks, control blocks. They are also given a pencil to fill in fields on the blocks, e.g. condition for `if statement`. There is no time limit, but the task is timed. Participant is not told that the number of failed executions they perform is recorded.

Roles and Experimental Method

We divided our team into these roles to conduct the experiment:

- A data entry worker to record the independent and dependent variables for each participant.
- Two facilitators to deliver a prepared explanation of the experiment for participants on each interface respectively. Firstly, they must ask how much experience the participant has with Scratch. Then they must explain what the blocks do and how to use them in a Scratch program.
- For each interface type, the facilitator should demonstrate connecting blocks. Finally they must explain the task the participants must perform, ask if there's anything that needs reiterating and confirm whether their code produces the correct output.
- Two timers to time the participants throughout the experiment. Start the timer after the facilitator's explanation, pause when green flag pressed, resume

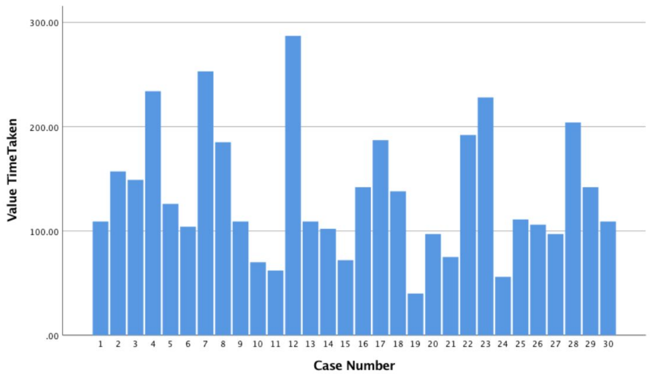

Figure 4 - Graph to show results of 30 participants of experiment

Group Statistics

	GroupType	N	Mean	Std. Deviation	Std. Error Mean
TimeTaken	Digital	15	145.5333	64.23714	16.58596
	Tangible	15	124.6000	59.73250	15.42286

Figure 5 - Table showing group statistics for data sets of tangible and digital

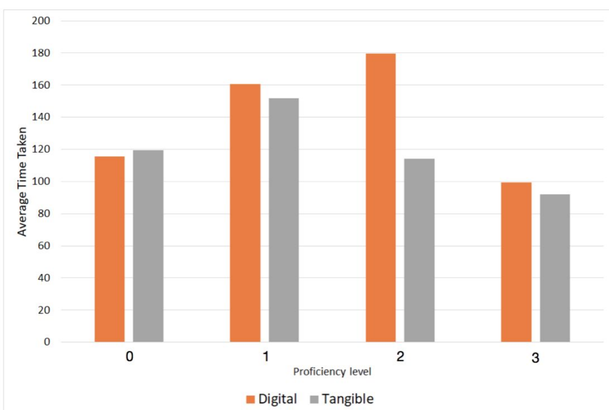

Figure 6 - Graph showing average time taken for each interface, at each proficiency level

the timer if the program fails and end timing on the first successful execution of the participant's program.

- A wizard to input the tangible interface program to a digital scratch interface on a hidden computer. When the green flag is pressed, shows the result of the program to the participant.

Results

Figure 4 shows the individual results of the 30 participants, as a time in seconds, of the experiment and includes results for both the tangible and digital tests. While there is a large range of results, none match the criteria required to remove them as an outlier as none are greater than, or less than, the mean plus or minus three times the standard deviation, the actual values for which can be seen in figure 5.

In figure 7, we can see the data compared in a histogram format. From this we can see that the data sets are relatively similar, with the modal time interval the same for both tangible and digital interfaces.

In figure 8, we can see there is very little difference in the mistakes made by users of the digital and tangible interfaces, suggesting very little impact is made in errors by switching from digital to tangible.

To see if we had proved our hypothesis with the results given above, we carried out T-Tests to check whether there was a significant enough difference between the two interfaces to come to a conclusion. First of all, we remove any outliers in the data, which are results that lie outside of 3 standard deviations away from the mean. However, as mentioned previously, none of our results are classed as outliers and therefore we can use the whole dataset for analysis. For there to be a significant enough difference to be able to prove or disprove our hypothesis, we would need a significance level of less than 0.05 so that there is less than a one in twenty chance of the results being due to randomness.

Looking at the time taken of the two groups, tangible and digital, regardless of proficiency in scratch we can see from the histogram that tangible does appear to be faster, however running a T-Test on the results we get a 0.363 p-value, which is not enough to reject the null hypothesis that there is no difference between tangible and digital. For mistakes made by the two groups, when comparing again using a T-Test, we get a p-value of 0.596, which means it appears even more likely that any differences in the mistakes made between the two interfaces is down to random chance.

Finally, running T-Tests on the time taken for experienced users, and new users separately, we get p-values of 0.157 and 0.954 respectively. While this isn't enough to reject the null hypothesis for experienced users, it does suggest there is a bigger difference for experienced users between the interfaces compared to new users, who show almost no difference in the results.

Limitations

There were several problems with the experiment that we noticed after carrying it out. The three main limitations to this experiment were:

The time taken to complete the task. These were inaccurate due to the timers being easily distracted by the environment. To prevent this, a future experiment could use electronic methods of timing so that the timer starts and stops as soon as the participant begins and stops coding.

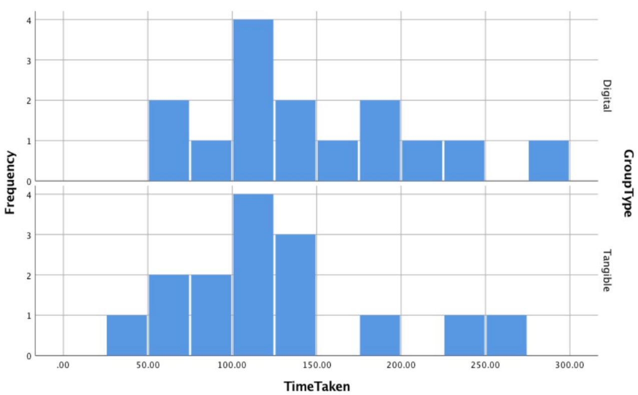

Figure 7 - Histogram comparing digital to tangible interfaces

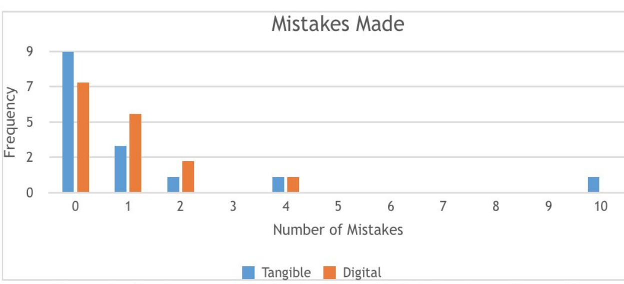

Figure 8 - Graph comparing mistakes made by digital and tangible participants

The method in which hints were given to participants. Due to the nature of the task, participants had to perform some mental arithmetic. There was an imbalance in how participants were helped with this, some being allowed to use calculators, others straight up being told the answer and others having no help at all. This did not produce a fair test and was a variable we did not control for. Therefore, in future there should be clear guidelines on when and how participants are given hints, or remove this problem completely by giving participants an easier task such as drawing a square instead of a pentagon.

The limited number of blocks in the tangible interface. The digital interface provided an unlimited number of blocks for the user to use. This was something we could not emulate on the tangible interface. This meant that the experience for users of the two interfaces was different as the digital interface participants could use a larger variety of blocks. To solve this, it would be best to limit the digital interface to the same quantity of each block as the tangible interface, as it would be unreasonable to have an 'unlimited' number of physical blocks.

Conclusion

After the statistical analysis of the experiment had been conducted, the following deductions came up. To begin with, it can be observed that the tangible interface was slightly faster than the digital. However, from the results of the statistical tests, we have only managed a p-value of 0.363 when comparing the two interfaces for all proficiencies. While this may suggest the tangible is slightly faster overall, it is not below our threshold of 0.05 required for significance, therefore we do not have sufficient data to reject the null hypothesis, and must conclude from our results that there is no difference in time taken to complete our task between the digital and tangible interfaces.

When looking at the other T-Tests we ran, we can see a large difference in p-values between experienced and new users, we cannot conclude that the tangible interface is faster for experienced users, as even though a p-value of 0.157 is lower than the p-value for all proficiencies, 0.363, it is still not below the 0.05 mark required for significance.

The similarity observed, can be attributed to the fact that the two interfaces are indeed similar in terms of speed, but also some other factors play a role, many of which contributed to the results being unreliable. First, the task assigned to the people who took part in the experiment required them to perform some arithmetic calculations, many of which struggled with it, and some encountered no problem. Thus, a problem with simpler calculations could have been used to avoid a scenario where some participants were at a disadvantage to start with. Secondly, the environment was quite distracting, and hence the people (participants and experimenters included) were not concentrating fully on the task; a quieter place could have been used to avoid this. Thirdly, In the tangible interface, only a finite number of blocks could have been used, implying an ease with respect to how these blocks can be combined. Finally, the people who took part in the experiment have various programming backgrounds, ranging from first year non-CS students to CS professors. Hence, a measure of the programming background could have been taken, instead of recording only their familiarity with Scratch.

Overall, there was not a significant enough difference in the results to be able to conclude decisively that one interface was faster than the other. Further experiments are needed with more variables being controlled to get an accurate assessment of which version of Scratch is fastest.

Team organization

Name	Contribution (%)
Callum Fawcett	14
Andreas Hadjiantonis	14
Liam Hunt	14
Kenny Lomas	14
Faizaan Sakib	14
Angus Williams	16
Felix Williams	14

Citations

[1] Primo Toys, Cubetto: A robot teaching kids code & computer programming, https://www.primotoys.com/ (2017)
[2] Physicad, Instructiblocks presented at TEI’17, http://physicad.com/2017/03/29/instructiblocks-presented-at-tei17/ (2017)
[3] Xu, D., Tangible User Interface for Children - An Overview. in Proceedings of the SIXTH Conference in the Department of Computing, (2005).