

# Taprova: Designing Tangible Representations of Variables in the Processing Language

IRIS BATAILLE, Department of Industrial Design, Eindhoven University of Technology, The Netherlands

ANNIEK JANSEN, Department of Industrial Design, Eindhoven University of Technology, The Netherlands

ANIKA KOK, Department of Industrial Design, Eindhoven University of Technology, The Netherlands



Fig. 1. Interacting with the Taprova tangibles.

Programming is becoming a more important skill to have. However, it can be challenging for novices to start developing this skill. Many educational programming tools have been developed already, but there is a lack of tangible tools that teach a specific programming language. This is why Taprova has been designed. Taprova consists of four tangible user interfaces that represent the variables String, integer, float and boolean of the Processing language. Both an experience prototype and a proof of concept prototype were made to explore the possibilities and to validate the design. Through a video-guided questionnaire, it was tested if the tangibles represent their digital version well. More than half of the participants understood the tangible representations, which makes Taprova a moderately effective tool for explaining the Processing variables. By exploring the use of tangible representations to teach a programming language, this research opens up a new sub-field within programming education for novices.

Additional Key Words and Phrases: programming education, tangible user interfaces, NFC

## ACM Reference Format:

Iris Bataille, Anniek Jansen, and Anika Kok. 2018. Taprova: Designing Tangible Representations of Variables in the Processing Language. In *UIST '21: ACM Symposium on User Interface Software and Technology, June 03–05, 2021, Woodstock, NY*. ACM, New York, NY, USA, 48 pages. <https://doi.org/10.1145/1122445.1122456>

## 1 INTRODUCTION

Programming is on the rise as a key and needed skill, which is why it is one of the 21st-century skills [14]. By learning how to program, other skills such as logical thinking and problem solving are taught as well [10]. However, programming challenges arise for novices due to the radical novelty of concepts and material and difficulty in understanding the syntax. This results in a mismatch between what programs do and what novices think they do, evoking strong negative feelings and creating barriers to learning [12].

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2018 Association for Computing Machinery.

Manuscript submitted to ACM

Since it is challenging for novices to learn how to program [17], several tools for teaching computational thinking have been developed [5, 6, 15, 19, 21–23]. To make programming easier, several educational programming games have been created. These all have similar features e.g. a playable character in a puzzle game made for single player use, controlled by a mouse and keyboard [12].

Programming skills can be influenced by the skills in computational thinking [18]. Most educational programming games that have been developed focused on teaching computational thinking, by teaching problem-solving strategies for example. In contrast, little tools that help to learn a specific programming language have been designed [4]. This study addresses this gap by exploring the opportunities for teaching the specific programming language Processing, rather than only computational thinking.

When teaching, Tangible User Interfaces (TUIs) have the potential to explain abstract concepts such as programming languages and concepts [24, 25]. Thanks to their hands-on nature, TUIs can offer more sensory engagement, are more accessible and facilitate group learning better than their Graphical User Interface (GUI) alternatives [11, 25]. Besides, research has shown that physicality has a large effect on teaching programming [17]. While TUIs are used to teach computational thinking [5, 6, 15, 19, 21–23], they are rarely used for teaching a programming language.

This study explores the opportunities for designing a TUI to teach the Processing language to novices, starting with declaring variables. For this purpose, Taprova was designed. Taprova stands for ‘Tangible Processing Variables’ and consists of four tangibles that each represent one variable of the Processing language. Experience prototypes and a proof of concept prototype of Taprova were made using an NFC array to define and test the design. The effects of the Taprova tool on the understanding of the Processing variables were studied through a video-guided questionnaire targeted to programming novices.

This research generated insights on the effects of Taprova on the understanding of variables in the Processing language. It was found that Taprova is a moderately successful tool, with just over half of the participants understanding the representations of the tangibles. By introducing a new tool for teaching a specific programming language, providing information about the effect of this tool and by discussing concrete suggestions for future work, this research contributes to the field of programming education for novices.

## 2 RELATED WORK

### 2.1 21st-century skills

Skills such as working with emerging technologies (e.g. AI and big data), critical thinking, leadership, creativity and technology, design and programming are becoming of increasing importance in the 21st century and are also called 21st-century skills[14]. To teach these skills to students, programming might be used and it is proposed to include programming in the K-12 curricula [16]. Programming can help to teach students computational thinking, and skills learned during programming can foster other skills such as creativity, logical thinking and problem-solving [10].

Most of the existing tools and literature focus on computational thinking [4, 18]. However, tools to teach a specific programming language are scarce. Besides the aforementioned potential benefits of learning to program, learning a programming language can be more appropriate for young adults, students and adults compared to only learning computational thinking. Mastering a programming language allows you more freedom in programming and increases the speed once you know the syntax since you do not have to search for the right visuals. Especially when using programming in a professional environment, this will be more useful.

## 2.2 Programming education

Learning to program is difficult for novices due to the radical novelty of concepts and material and difficulty in understanding the syntax [12]. This results in a mismatch between what programs do and what novices think they do, evoking strong negative feelings and creating barriers to learning. Furthermore, novices do not only need to learn how to program but also how to troubleshoot, self-reflect and organize a code which increases the level of difficulty [17].

To make programming easier and to address the growing needs of computer science education, several educational programming games have been created [12]. The majority of these educational programming games consist of: playable characters (e.g. robot) in the puzzle genre, with the same mode of play (e.g. single player) and with similar input methods (e.g. a touchpad or a mouse and keyboard).

*Existing GUI tools.* Visualization has proven to have a moderate effect on teaching programming [18]. Since visualization reduces the cognitive load which makes it more accessible and visualizations can help to create mental models. However, visuals can also be distracting and do not use the body awareness and skills of the user [8]. Several GUI tools are developed and they often use coloured blocks to represent the different commands and functions [1–3, 13]. However, visuals can also be distracting and do not exploit the bodily awareness skills of the user [8].

*Existing TUI tools.* Tangibles fully exploit the naive physics bodily awareness skills of the user [8] and the effect of physicality is large on teaching programming [17]. Moreover, TUIs have the potential to support learning and collaboration, increase the performance (e.g. fewer mistakes), enhance memory and be more engaging [25]. However, these advantages remain mostly theoretical and empirical evidence is inconclusive [11]. Nevertheless, many tangible tools are made to support teaching computational thinking [5, 6, 15, 19, 21–23]. This study aims to explore the use of tangibles to teach novices how to declare variables in the Processing language.

## 3 DESIGN

For this research, Taprova was designed. Taprova consists of four tangible blocks with different interaction possibilities (see figure 2). Since this research targets novices, it was decided to focus on one aspect of programming: variables. The four interfaces each represent one variable of the Processing language: String, integer, float and boolean. These four were chosen because of their use frequency during the programming process. To create consistency and coherence, the interfaces have a similar shape and size but are modified to represent the variable. By taking into account the naive physics and body awareness and skills that people have [8], each interface was designed so that its affordances match the variable. The String is represented by a frame in which physical characters can be placed. The integer is represented

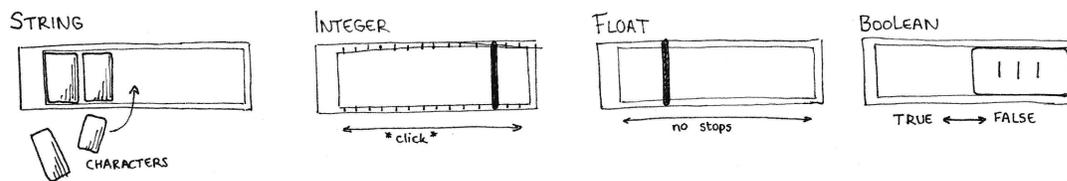


Fig. 2. A sketch of the physical objects that represent the variables String, integer, float and boolean

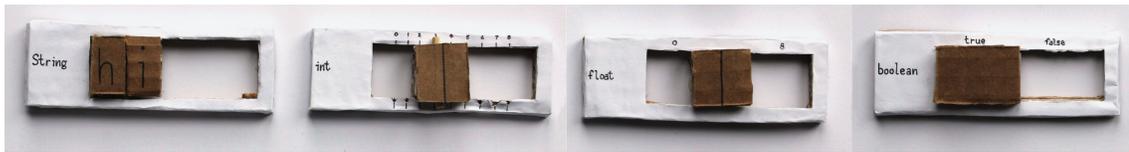


Fig. 3. The experience prototypes of Taprova.

by a slider that snaps at a whole number. The float is represented by a slider that has no stops. Lastly, the boolean is represented by a switch.

Two different types of prototypes were constructed. The first are experience prototypes made of cardboard (see figure 3). These prototypes were used to show the interaction with the interfaces in videos which were used in the video-guided questionnaire (see Method). Next to that, a proof of concept prototype was made using NFC technology. This technology was selected because it does not require any power supply or microcontroller in the tangibles, and it does not depend on the tangibles being in the line of sight which would be the case when using camera recognition.

An MFRC522 RFID reader can only recognize one tag when used in its original state. To detect more tags the reader is customized (see figure 4a) [9], by cutting the circuit traces on the reader next to the on-board capacitors. One side of the traces is connected to an HC4067 16 channel analog multiplexer. The other side connects in parallel to one leg of eight customized antennas cut from copper tape with 5 turns and a dimension of 40\*40 mm with circuit traces of 1.2 mm, this results in a similar conductance as the original antenna. The other leg of the antennas is connected to the input pins of the multiplexer. The traces overlap each other to prevent blind spots. Each antenna can detect one tag, resulting in a maximum recognition of 8 tags.

Two Mifare classic 1K tags were connected to each tangible, one on a stationary location and one on the slider (see figure 4b). The distance between them translated in the value for the integer and float. The String used the tags to recognize the letters and their position and the value of the boolean depended on whether the slider was over a preset threshold to create a switch effect.

An Arduino Uno is used with the SPI.h, RFID.cpp and RFID.h libraries to loop through the antennas and check for the tags. Through serial communication, these readings are sent to Processing<sup>1</sup> which identified the tags as stationary or slider and calculated the distance and value based on its current mode (String, float, integer or Boolean). This value was displayed in a screen resembling the normal Processing interface (see figure 4c) and was live updated depending on the location of the slider and the presence of the tags. The declaration of the variable was only shown when a tag was present. A video of the use of the proof of concept prototype can be found at <https://youtu.be/V8iAhHITMi8>.

#### 4 METHOD

Before testing the effectiveness of the entire tangible toolkit, the individual variables were researched to test whether the tangibles as presented in chapter 3, represent the digital Processing variables they were designed to physicalize. The research question was, therefore: *Are the tangible objects of Taprova an accurate representation of the digital Processing variables that they try to convey?*. To validate the ideas and to get preliminary results on how the tangibles influence the programming skills of novices, a video-guided questionnaire was used. This way, the effects of the tangibles could be tested in a safe, remote way complying with the COVID-19 safety regulations that were active at the time of researching. Participants were selected using snowball sampling. The target group consisted of programming novices between the

<sup>1</sup><https://processing.org/>

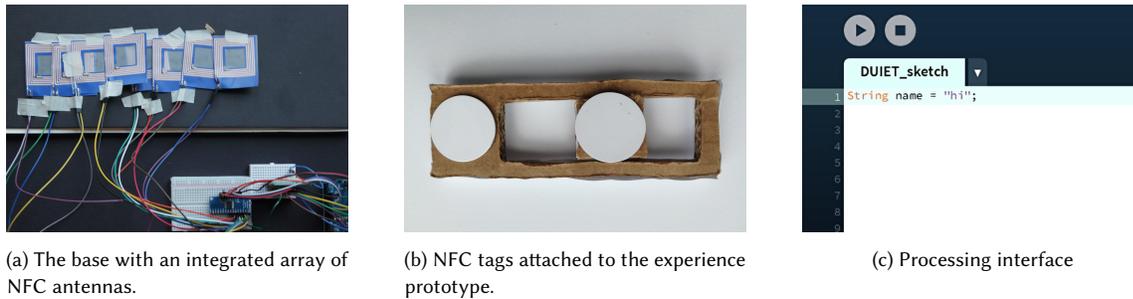


Fig. 4. The different elements of the proof of concept prototype.

ages of 18-70, with no further requirements to in- or exclude participants from the research. The questionnaire was available for four days in which 37 questionnaires were completed.

*Procedure.* Through the videos in the questionnaire, four tangible objects designed to visualise digital Processing variables were presented to the participants. Based on these videos, participants were able to form a personal interpretation of the meaning of each variable, after which they were asked to evaluate each of the variables both individually and in combination with the digital Processing programming environment.

In the questionnaire, quantitative data were collected primarily using 2 Likert scale questions, 14 open questions and 21 multiple choice questions. The questionnaire contained multiple sections: (I) consent, (II) a questionnaire introduction, (III) Demographic information, (IV) evaluation of each variable, (V) evaluation of each variable combined with Processing interface, (VI) summary questions and (VII) final questions. The complete survey can be found in appendix A.

The order in which the variables were presented was randomised to ensure the validity of the results. The questions and multiple choice answers in sections IV and V were identical. The only difference being the video, where participants were first presented with a video of only the tangible and secondly with a video of the tangible combined with the Processing graphical user interface. The playlist containing the used videos can be opened with this link: <https://youtube.com/playlist?list=PLz9hncultkrGTMSPAeraIo2j34sbfddM>. An example of questions used in these sections can be seen in Table 1.

*Analysis.* The data was cleaned in a Jupyter Notebook by removing all participants who answered five or more times "I don't know" (6 participants) to the description questions after the videos or who answered "Yes" to the questions if they ever programmed before (6 participants). In total, the data from twelve participants were removed. The multiple-choice questions from the questionnaire were coded as 'correct', 'incorrect', and 'I don't know'. Each variable had two questions after each video, resulting in 8 questions per participant per phase for phase IV and V. The total for each option was counted per participant after the first video and the second video. A second database was made with the answers filtered by variable and participant. Table 2 shows both the descriptive statistics by participant as well as the total number of correct, incorrect and I don't know answers for each variable (50 answers in total). The distribution of the differences was analysed for each answer option using a boxplot. The distribution was not symmetrical and hence violated the assumption for using a Wilcoxon-Signed Rank Test. Instead, the Sign Test was used for analysing both the overall results and the results for each variable. Dummy variables were created for the nominal variables: age and interesting programming. These were used in two multiple linear regressions to predict if they influence the number of correct

Table 1. Example questions as used in the questionnaire

Question	Option 1	Option 2	Option 3	Option 4	Option 5
What do you think is the definition of a String?	A whole number	A decimal number	True or False	A sequence of characters	I don't know
What do you think is an example of a String?	true	3	"true"	3.0	I don't know

answers after the second video and the improvement between the first and second results. All analyses were carried out in the SPSS software (version 25) [7].

## 5 RESULTS

None of the participants was able to declare a String in the Processing language at the beginning of the questionnaire. The mean time spent on completing the questionnaire was 13 minutes and 40 seconds (excluding one outlier of 5 hours and 34 minutes).

*Improvement during the test.* The descriptive statistics in Table 2 show that the number of correct answers increased after seeing the second set of videos which included both the tangible variables and the Processing interface; the number of 'I don't know' decreased and the number of wrong answers stayed roughly the same. After the first set of videos, 41.3% of the questions were answered correctly which increased to 55.0% after the second set of videos. A Sign Test showed that these differences are not significant at a 95% confidence level (correct answers:  $p=0.180$ , I don't know answers:  $p=0.092$  and wrong answers:  $p=1.000$ ).

A minority of the participants (12%) was able to recognize the name and type of the variable from a line of code at the end of the questionnaire and 32% was able to correctly identify the value. However, none of the participants was able to declare a String at the end of the questionnaire.

*Variable types.* To compare the first and second test results of each variable, a sign test was used four times. While the number of correct answers increased for each variable this difference was only significant for the float ( $p=0.039$ ). No significant differences were found for the number of incorrect and I don't know answers.

*Influence of age and interest.* A multiple linear regression was calculated to predict the number of correct answers based on age and interest in programming. No significant regression equation was found ( $F(8,16)=0.650, p=0.726$ ) with an  $R^2$  of 0.245. A second multiple linear regression was calculated to predict the improvement the participants made between the first set of videos and the second set based on their age and interest in programming. No significant regression equation was found ( $F(8,16)=1.961, p=0.120$  with an  $R^2$  of 0.495).

## 6 DISCUSSION

The results of the questionnaire show that the number of correct answers regarding the tangible Processing variables is not fully convincing. The combination of 'I don't know' or wrong answers selected by participants in the first round of

Table 2. Descriptive statistics

	Sorted by participant					Total number for each variable type			
	N	Mean	SD	Min	Max	Integer	Float	Boolean	String
First results Correct	25	3.32	2.61	0	8	18	17	29	19
First results Incorrect	25	1.40	1.63	0	5	15	8	4	8
First Results I don't know	25	3.32	2.95	0	8	17	25	17	23
Second results Correct	25	4.40	2.97	0	8	27	30	31	22
Second results Incorrect	25	1.24	1.67	0	6	11	4	8	8
Second results I don't know	25	2.36	3.09	0	8	12	16	11	20
Programming interest (1-5)	25	2.04	1.14	1	5				
Interest tangibles (1-5)	25	2.56	1.50	1	5				

introducing the variables through the Taprova designs was more than the number of correct answers. This indicates that just the design and the interaction as shown in the video are not enough for participants to immediately understand the meaning of the Processing variables. Only in case of the boolean variable, participants selected the correct answer notably more often than the 'I don't know' or incorrect options. This suggests that the boolean is the only variable that was better understood by participants based on the information presented in the questionnaire.

However, once participants were shown the Taprova designs in combination with the Processing user interface, the number of correct answers increased slightly, leaving the number of selected incorrect and 'I don't know' options lower than before. These results indicate that the combination of the Taprova tangible and the Processing interface provide clearer information from which the meaning of the variables can be deduced. It is also possible that the improvements come from the learning effect that is happening over the course of the questionnaire.

Based on the fact that none of the participants had programming experience, and none of them was able to declare a String correctly at the start of the research, a score of 55% correct answers suggests that Taprova is a moderately successful tool to teach novices how to understand variables in the Processing language. The improvement was only visible in the multiple-choice questions which indicates that participants are not able to recall the knowledge in open questions that require a better understanding. An explanation for this can be the short time frame in which participants were introduced to the tangible variables. Better results may be obtained after a longer period of seeing and/or working with the tangibles. Also, not being able to interact with the tangibles personally might have affected the understanding of the variables.

When asked whether they would want to try out the tangibles in real life, half of the participants showed interest and half of the participants did not. This indicates that participants are not motivated to keep on learning by means of the Taprova tangibles. A possible explanation could be the intrinsic motivation, which is low for people that are not interested in programming. It could also mean that the current design does not add enough value for participants to convince and inspire them to keep learning.

Future research should be conducted to validate the conclusions as stated and to study the effects of Taprova further. Think of tests in which the participants can actually use the Taprova tangibles themselves physically and/or studies that take place over a longer period of time. Moreover, tangibles have been proven to be better for collaboration and collaboration results in better scores [16]. A comparative study could be conducted to investigate if Taprova is suited for this purpose.

## 6.1 Limitations

*Study limitations.* A few limitations of the user study as performed can be noted. Firstly, the sample size of the study was limited. A larger sample would have resulted in more valid results and might have a symmetrical distribution of the differences, allowing for a more powerful one-tailed Wilcoxon Signed Rank test to analyse the results.

Besides the small sample size, the findings might be influenced by the fact that the questionnaire itself could be seen as a learning tool. By presenting the variables in random order, this effect was minimized. Also, most of the questions regarding the variables were multiple-choice, allowing for the possibility of guessing correctly. By adding the option: 'I don't know', an attempt was made to minimize this effect.

Lastly, studying the effects of Taprova in an online questionnaire causes the impossibility for participants to personally interact with the variables. This prohibited them from actually experiencing the physical interaction and the effects of this interaction on the outcome in the Processing GUI. Therefore, this can be counted as a limitation of the study.

*Design limitations.* The first design limitation relates to the constraints that physicality entails in this case. In its current design, the output of the design is constrained to a limited amount of characters or a limited number for example. Really long words or high numbers cannot be created in the current form of Taprova.

The second design limitation lies in the resolution of the NFC-array. The Taprova design currently uses an NFC-array to locate the variables and read the values. However, the resolution of this array is too low to get accurate values for the float and the tags are relatively large for the characters. Using the mat from Project Zanzibar [20] would solve the problem of low resolution and the large NFC tags since it uses multi capacitance touch sensing which offers detection of higher resolution. This technology also opens the opportunity to incorporate touch into the design of the tangibles which could be used to connect variables, reverse order or change the range of a variable inside the tangibles.

## 7 CONCLUSION

This research aimed to design tangible representations, Taprova, of the variables integer, float, boolean and String in the Processing language and to evaluate if they are accurate representations of the digital variables. The results of a video-guided questionnaire showed that 55% of the answers were answered correctly after seeing two sets of videos, making Taprova a moderately effective representation. The results after seeing a video of Taprova did not differ significantly from the results after seeing a video of both Taprova and the Processing interface, except for the float variable where the number of correct answers increased significantly. The results of this study suggest that tangible representations can be made to teach programming languages to novices which opens a new research area related to tangible programming education. Future research is needed to test if our findings can be extended to a situation where participants interact live with the tangibles and to investigate what the effects are when interacting for a longer period with the tangibles.

## ACKNOWLEDGMENTS

We would like to thank Rong-Hao Liang, Zengrong Guo, Laura Bijl and Noa van der Horst for their support and the provided materials for realizing the prototypes.

## REFERENCES

- [1] Ardublock. [n.d.]. *Ardublock A Graphical Programming Language for Arduino*. <http://blog.ardublock.com/>
- [2] Julián da Silva Gillig. [n.d.]. *miniBloq is an open source graphical programming environment for Multiplo™, Arduino™, physical computing devices and robots*. <http://blog.minibloq.org/>

- [3] EduBlocks. [n.d.]. *BBC micro:bit*. <https://edublocks.org/microbit.html>
- [4] Julie Henry, Antoine Bodart, and Bruno Dumas. 2018. Tangible Programming for Children: State of the Art, Classification and Opportunities. In *Proceedings of the 30th Conference on l'Interaction Homme-Machine* (Brest, France) (*IHM '18*). Association for Computing Machinery, New York, NY, USA, 241–249. <https://doi.org/10.1145/3286689.3286711>
- [5] Michael S. Horn and Robert J. K. Jacob. 2006. Tangible Programming in the Classroom: A Practical Approach. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems* (Montréal, Québec, Canada) (*CHI EA '06*). Association for Computing Machinery, New York, NY, USA, 869–874. <https://doi.org/10.1145/1125451.1125621>
- [6] Michael S. Horn and Robert J. K. Jacob. 2007. Tangible Programming in the Classroom with Tern. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems* (San Jose, CA, USA) (*CHI EA '07*). Association for Computing Machinery, New York, NY, USA, 1965–1970. <https://doi.org/10.1145/1240866.1240933>
- [7] IBM Corp. [n.d.]. *IBM SPSS Statistics for Windows*. Armonk, NY: IBM Corp. <https://hadoop.apache.org>
- [8] Robert J.K. Jacob, Audrey Girouard, Leanne M. Hirshfield, Michael S. Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. 2008. Reality-Based Interaction: A Framework for Post-WIMP Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) (*CHI '08*). Association for Computing Machinery, New York, NY, USA, 201–210. <https://doi.org/10.1145/1357054.1357089>
- [9] Rong-Hao Liang. [n.d.]. Customizing NFC Antenna. <https://paper.dropbox.com/doc/Customizing-NFC-Antenna--BD3bwlelQMj1-MBa8i8tSvJnAg-S62egA3ZsAaCoyp1Wqtd8>
- [10] Yuen-Kuang Cliff Liao and George W Bright. 1991. Effects of computer programming on cognitive outcomes: A meta-analysis. *Journal of Educational Computing Research* 7, 3 (1991), 251–268.
- [11] Paul Marshall, Sara Price, and Yvonne Rogers. 2003. Conceptualising Tangibles to Support Learning. In *Proceedings of the 2003 Conference on Interaction Design and Children* (Preston, England) (*IDC '03*). Association for Computing Machinery, New York, NY, USA, 101–109. <https://doi.org/10.1145/953536.953551>
- [12] Edward F. Melcer and Katherine Isbister. 2018. Bots & (Main)Frames: Exploring the Impact of Tangible Blocks and Collaborative Play in an Educational Programming Game. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (*CHI '18*). Association for Computing Machinery, New York, NY, USA, 1–14. <https://doi.org/10.1145/3173574.3173840>
- [13] MIT. [n.d.]. *Imagine, Program, Share*. <https://scratch.mit.edu/>
- [14] Stephen K Reed. 2020. *Cognitive Skills You Need for the 21st Century*. Oxford University Press, USA.
- [15] Zhiyi Rong, Ngo fung Chan, Taizhou Chen, and Kening Zhu. 2020. CodeRhythm: A Tangible Programming Toolkit for Visually Impaired Students. In *The Eighth International Workshop of Chinese CHI* (Honolulu, HI, USA) (*Chinese CHI 2020*). Association for Computing Machinery, New York, NY, USA, 57–60. <https://doi.org/10.1145/3403676.3403683>
- [16] Ronny Scherer. 2016. Learning from the past—the need for empirical evidence on the transfer effects of computer programming skills. *Frontiers in psychology* 7 (2016), 1390.
- [17] Ronny Scherer, Fazilat Siddiq, and Bárbara Sánchez Viveros. 2019. The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology* 111, 5 (2019), 764.
- [18] Ronny Scherer, Fazilat Siddiq, and Bárbara Sánchez Viveros. 2020. A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions. *Computers in Human Behavior* (2020), 106349.
- [19] Andrew Cyrus Smith. 2009. Symbols for Children’s Tangible Programming Cubes: An Explorative Study. In *Proceedings of the 2009 Annual Conference of the Southern African Computer Lecturers’ Association* (Eastern Cape, South Africa) (*SACLA '09*). Association for Computing Machinery, New York, NY, USA, 105–109. <https://doi.org/10.1145/1562741.1562755>
- [20] Nicolas Villar, Daniel Cletheroe, Greg Saul, Christian Holz, Tim Regan, Oscar Salandin, Misha Sra, Hui-Shyong Yeo, William Field, and Haiyan Zhang. 2018. Project Zanzibar: A Portable and Flexible Tangible Interaction Platform. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (*CHI '18*). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3173574.3174089>
- [21] Danli Wang, Yunfeng Qi, and Lan Zhang. 2015. A Tangible Programming System Conveying Event Handling Concept. In *Proceedings of the 14th International Conference on Interaction Design and Children* (Boston, Massachusetts) (*IDC '15*). Association for Computing Machinery, New York, NY, USA, 319–322. <https://doi.org/10.1145/2771839.2771906>
- [22] Danli Wang, Yang Zhang, Tianyuan Gu, Liang He, and Hongan Wang. 2012. E-Block: A Tangible Programming Tool for Children. In *Adjunct Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) (*UIST Adjunct Proceedings '12*). Association for Computing Machinery, New York, NY, USA, 71–72. <https://doi.org/10.1145/2380296.2380327>
- [23] Peta Wyeth and Helen C. Purchase. 2002. Tangible Programming Elements for Young Children. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems* (Minneapolis, Minnesota, USA) (*CHI EA '02*). Association for Computing Machinery, New York, NY, USA, 774–775. <https://doi.org/10.1145/506443.506591>
- [24] Oren Zuckerman, Saeed Arida, and Mitchel Resnick. 2005. Extending Tangible Interfaces for Education: Digital Montessori-Inspired Manipulatives. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) (*CHI '05*). Association for Computing Machinery, New York, NY, USA, 859–868. <https://doi.org/10.1145/1054972.1055093>
- [25] Oren Zuckerman and Ayelet Gal-Oz. 2013. To TUI or not to TUI: Evaluating performance and preference in tangible vs. graphical user interfaces. *International Journal of Human-Computer Studies* 71, 7 (2013), 803 – 820. <https://doi.org/10.1016/j.ijhcs.2013.04.003>

## **A QUESTIONNAIRE QUESTIONS**

# Programming with tangible objects

Thank you for taking the time to fill in this questionnaire. It should only take up to ten minutes of your time!

The questionnaire is completely anonymous. Your participation is completely voluntary and you are free to stop at any time. If you want to contact the researchers for for instance more information about this study, the study design, or the results, you can contact one of the researchers:

- Iris Bataille (i.h.a.bataille@student.tue.nl),
- Anniek Jansen (a.jansen1@student.tue.nl),
- Anika Kok (a.k.kok@student.tue.nl).

If you have any complaints about this study, please contact the supervisor, Rong-Hao Liang (j.liang@tue.nl). You can report irregularities related to scientific integrity to confidential advisors of the TU/e.

This project is part of a research project at the Eindhoven University of Technology. All the data will be securely stored and will be used for research purposes only. All information will be handled anonymously and confidential, in accordance with the ethical guidelines defined by the University. If you agree please tick the box and continue to the survey.

\* Required

## Consent

1

After reading the survey description please select which applies to you. \*

- I agree with the above stated terms and wish to continue with this survey.
- I do not agree with the above stated terms and wish to discontinue this survey. Choosing this option will end the survey.

## Questionnaire introduction

This questionnaire will be about programming, specifically in the programming language Processing. In this questionnaire you will be shown multiple videos. Each of the videos visualize a tangible object that represents a variable of the coding language Processing. With this tangible, you can adjust the value of the represented variable.

Please watch each video carefully and answer the questions afterwards.

2

Have you ever programmed before? \*

Yes

No

4

Please declare a string in the Processing language

*You can leave this question empty if you don't know.*

## Demographic information

5

How old are you? \*

- Younger than 20
- Between 21 and 30
- Between 31 and 40
- Between 41 and 50
- Between 51 and 60
- Between 61 and 70
- Between 71 and 80

6

How do you identify your gender? \*

- Woman
- Prefer not to say
- Man
- Genderqueer/Non-binary

Other

7

In what quarter of the year is your birthday? \*

- Quarter 1 (jan/feb/mar)
- Quarter 2 (apr/may/jun)
- Quarter 3 (jul/aug/sept)
- Quarter 4 (oct/nov/dec)

## | Variable: integer

Please watch the video before answering the questions.

8

Please describe what is happening in the integer video \*

9

What do you think is the definition of an integer? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

10

What do you think is an example of an integer? \*

- true
- 3
- "true"
- 3.0
- I don't know

## | Variable: integer with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

11

Please describe what is happening in the integer video. \*

12

What do you think is the definition of an integer?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

13

What do you think is an example of an integer?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## || Variable: integer

Please watch the video before answering the questions.

14

Please describe what is happening in the integer video \*

15

What do you think is the definition of an integer? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

16

What do you think is an example of an integer? \*

- true
- 3
- "true"
- 3.0
- I don't know

## || Variable: integer with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

17

Please describe what is happening in the integer video. \*

18

What do you think is the definition of an integer?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

19

What do you think is an example of an integer?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## ||| Variable: integer

Please watch the video before answering the questions.

20

Please describe what is happening in the integer video \*

21

What do you think is the definition of an integer? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

22

What do you think is an example of an integer? \*

- true
- 3
- "true"
- 3.0
- I don't know

## ||| Variable: integer with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

23

Please describe what is happening in the video. \*

24

What do you think is the definition of an integer?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

25

What do you think is an example of an integer?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## ||| Variable: integer

Please watch the video before answering the questions.

26

Please describe what is happening in the integer video \*

27

What do you think is the definition of an integer? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

28

What do you think is an example of an integer? \*

- true
- 3
- "true"
- 3.0
- I don't know

## ||| Variable: integer with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

29

Please describe what is happening in the integer video. \*

30

What do you think is the definition of an integer?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

31

What do you think is an example of an integer?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## | Variable: string

Please watch the video before answering the questions.

32

Please describe what is happening in the String video \*

33

What do you think is the definition of a string? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

34

What do you think is an example of a string? \*

- true
- 3
- "true"
- 3.0
- I don't know

## | Variable: string with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

35

Please describe what is happening in the String video. \*

36

What do you think is the definition of a string?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

37

What do you think is an example of a string?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## || Variable: string

Please watch the video before answering the questions.

38

Please describe what is happening in the String video \*

39

What do you think is the definition of a string? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

40

What do you think is an example of a string? \*

- true
- 3
- "true"
- 3.0
- I don't know

## || Variable: string with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

41

Please describe what is happening in the String video. \*

42

What do you think is the definition of a string?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

43

What do you think is an example of a string?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## ||| Variable: string

Please watch the video before answering the questions.

44

Please describe what is happening in the String video \*

45

What do you think is the definition of a string? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

46

What do you think is an example of a string? \*

- true
- 3
- "true"
- 3.0
- I don't know

## ||| Variable: string with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

47

Please describe what is happening in the String video. \*

48

What do you think is the definition of a string?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

49

What do you think is an example of a string?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## ||| Variable: string

Please watch the video before answering the questions.

50

Please describe what is happening in the String video \*

51

What do you think is the definition of a string? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

52

What do you think is an example of a string? \*

- true
- 3
- "true"
- 3.0
- I don't know

## ||| Variable: string with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

53

Please describe what is happening in the String video. \*

54

What do you think is the definition of a string?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

55

What do you think is an example of a string?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## | Variable: boolean

Please watch the video before answering the questions.

56

Please describe what is happening in the boolean video \*

57

What do you think is the definition of a boolean? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

58

What do you think is an example of a boolean? \*

- true
- 3
- "true"
- 3.0
- I don't know

## | Variable: boolean with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

59

Please describe what is happening in the boolean video. \*

60

What do you think is the definition of a boolean?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

61

What do you think is an example of a boolean?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## || Variable: boolean

Please watch the video before answering the questions.

62

Please describe what is happening in the boolean video \*

63

What do you think is the definition of a boolean? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

64

What do you think is an example of a boolean? \*

- true
- 3
- "true"
- 3.0
- I don't know

## || Variable: boolean with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

65

Please describe what is happening in the boolean video. \*

66

What do you think is the definition of a boolean?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

67

What do you think is an example of a boolean?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## ||| Variable: boolean

Please watch the video before answering the questions.

68

Please describe what is happening in the boolean video \*

69

What do you think is the definition of a boolean? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

70

What do you think is an example of a boolean? \*

- true
- 3
- "true"
- 3.0
- I don't know

## ||| Variable: boolean with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

71

Please describe what is happening in the boolean video. \*

72

What do you think is the definition of a boolean?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

73

What do you think is an example of a boolean?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## |||| Variable: boolean

Please watch the video before answering the questions.

74

Please describe what is happening in the boolean video \*

75

What do you think is the definition of a boolean? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

76

What do you think is an example of a boolean? \*

- true
- 3
- "true"
- 3.0
- I don't know

## ||| Variable: boolean with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

77

Please describe what is happening in the boolean video. \*

78

What do you think is the definition of a boolean?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

79

What do you think is an example of a boolean?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## | Variable: float

Please watch the video before answering the questions.

80

Please describe what is happening in the float video \*

81

What do you think is the definition of a float? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

82

What do you think is an example of a float? \*

- true
- 3
- "true"
- 3.0
- I don't know

## | Variable: float with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

83

Please describe what is happening in the float video. \*

84

What do you think is the definition of a float?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

85

What do you think is an example of a float?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## || Variable: float

Please watch the video before answering the questions.

86

Please describe what is happening in the float video \*

87

What do you think is the definition of a float? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

88

What do you think is an example of a float? \*

- true
- 3
- "true"
- 3.0
- I don't know

## || Variable: float with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

89

Please describe what is happening in the float video. \*

90

What do you think is the definition of a float?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

91

What do you think is an example of a float?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## ||| Variable: float

Please watch the video before answering the questions.

92

Please describe what is happening in the float video \*

93

What do you think is the definition of a float? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

94

What do you think is an example of a float? \*

- true
- 3
- "true"
- 3.0
- I don't know

## ||| Variable: float with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

95

Please describe what is happening in the float video. \*

96

What do you think is the definition of a float?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

97

What do you think is an example of a float?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## ||| Variable: float

Please watch the video before answering the questions.

98

Please describe what is happening in the float video \*

99

What do you think is the definition of a float? \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

100

What do you think is an example of a float? \*

- true
- 3
- "true"
- 3.0
- I don't know

## ||| Variable: float with Processing

Please watch the video before answering the questions. This is the same video as shown before, but with the Processing code editor on the right.

101

Please describe what is happening in the float video. \*

102

What do you think is the definition of a float?. \*

- A whole number
- A decimal number
- True or False
- A sequence of characters
- I don't know

103

What do you think is an example of a float?. \*

- true
- 3
- "true"
- 3.0
- I don't know

## Summary questions

Fill in the blanks: The variable [name] is an [type] with value [value].



104

Look at the code in the image and the sentence with the blanks. What should be in the place of [name]? You can leave this question empty if you don't know.

*The variable [name] is an [type] with value [value].*

105

Look at the code in the image and the sentence with the blanks. What should be in the place of [type]? You can leave this question empty if you don't know.

*The variable [name] is an [type] with value [value].*

106

Look at the code in the image and the sentence with the blanks. What should be in the place of [value]? You can leave this question empty if you don't know.

*The variable [name] is an [type] with value [value].*

## Final questions

107

**|| Please declare a string in the Processing language.**

*You can leave this question empty if you don't know.*

109

**Do you have any other remarks? If not, leave the text box empty.**

*Thank you for taking the time to fill in our survey. Please use the text box down below to give any other remarks on for example the reasoning behind your answers or on the questionnaire itself. Leave the text box empty if you have no remarks.*

---

This content is neither created nor endorsed by Microsoft. The data you submit will be sent to the form owner.

Microsoft Forms

1/25/2021