Mathematics Input for Educational Applications in Virtual Reality

Conference Paper

Author(s):
Sansonetti, Luigi; Chatain, Julia; Caldeira, Pedro; Fayolle, Violaine; Kapur, Manu; Sumner, Robert W.

Publication date:
2021

Permanent link:
https://doi.org/10.3929/ethz-b-000514560

Originally published in:
https://doi.org/10.2312/egve.20211324

International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2021)
J. Orlosky, D. Reiners, and B. Weyers (Editors)

Mathematics Input for Educational Applications in Virtual Reality

Luigi Sansonetti[1], Julia Chatain[1,2], Pedro Caldeira[2], Violaine Fayolle[1], Prof. Dr. Manu Kapur[2], Prof. Dr. Robert W. Sumner[1]

[1]Department of Computer Science, ETH Zurich, Switzerland
[2]Department of Humanities, Social and Political Sciences, ETH Zurich, Switzerland

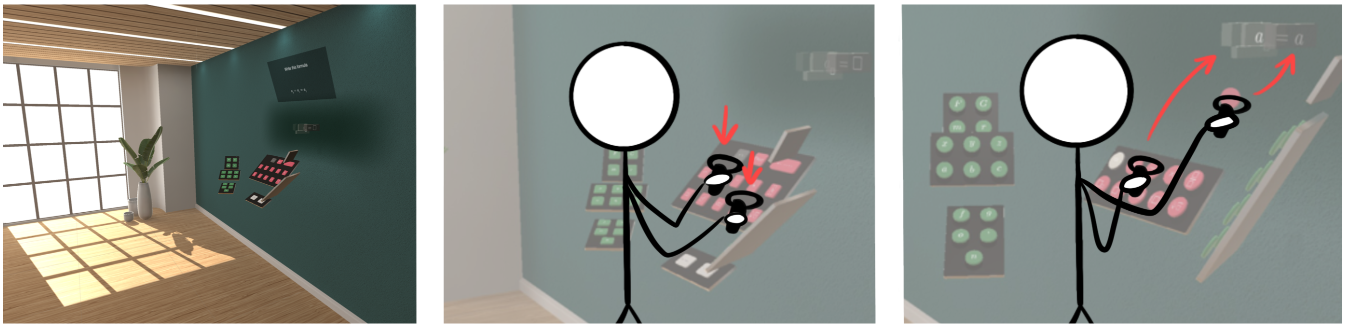


Figure 1: *The Virtual Reality workspace (left) and two Mathematics Input interfaces: a Keyboard-like (*KBD*) interface (center), and a Drag-and-drop (*DND*) interface (right)*

Abstract
*Virtual Reality (*VR*) enables new ways of learning by providing an interactive environment to learn through failure and by allowing new interaction methods engaging the users' bodies. Literature from productive failure and embodied cognition shows that these two aspects are particularly important for mathematics education. However, very little research has been looking into how to input mathematical expressions in* VR*. This gap impairs the learning process as it prevents the learners from connecting the* VR *mathematical objects with their formal representations. In this paper, we bridge this gap by presenting two interaction techniques for mathematics input in* VR*: a Keyboard-like method and a Drag-and-drop method. We report the results of our quantitative user study in terms of usability, ease of learning, low overhead, task load, and motion sickness.*

CCS Concepts
• ***Human-centered computing*** → ***Interaction techniques; Empirical studies in HCI; User studies;*** *Text input;* ***Virtual reality;***

1. Introduction

Modern technologies enable new ways of learning mathematics. Virtual Reality (VR) in particular, has a strong potential as it lets students learn through failure without requiring an immediate understanding of abstract concepts [Bri90]. Moreover, VR supports embodied interaction. Bringing bodies back at the core of the digital learning experience is a crucial aspect of education that is often left aside in traditional practices [MP14, ANWP*20, Spi21]. Embodied interaction allows to convey notions that students cannot yet describe with words or formal symbol systems [Rot01]. Additionally, involving the bodies of the users can alleviate cognitive load, enabling them to focus their cognitive resources on solving the problem at hand [TSB17].

Supporting the theory, empirical studies on VR educational applications report an increased interest in the topic being taught among participants [KSW00, SSE18]. However, although mathematical education in VR was well-received by students, a major obstacle to its wide-spread adoption is a lack of standard conventions for mathematical input. This is unfortunate as it prevents the students from actively reconnecting their VR mathematical objects with their corresponding formal representations.

We delve into this topic by implementing and evaluating two types of interaction to input mathematical expressions, namely a) a keyboard-like interaction, and b) a drag-and-drop interaction, both represented in Figure 1. In order to increase the sense of embodiment felt in VR [KGS12], both interaction methods require the users to interact with their environment to input symbols and expressions. Additionally, the second interaction provides a mnemonic mapping between the *act* of reaching for a certain element and the *idea* of bringing it into the expression. We evaluated our interaction techniques with a quantitative user study, focusing on performance, usability, motion sickness, and task load. We combine this analysis with participants' observations and provide a qualitative interpretation of the results.

2. Related Work

Previous research investigated how to input text in VR [DA19]. Speicher et al. offer six different methods for text entry in VR: head pointing, controller pointing, controller tapping, freehand, and discrete and continuous cursor [SFZK18]. Their research shows that the controller pointing method, that is, pointing at the characters with handheld controllers, outperformed other methods in terms of performance. Moreover, this method was the only method scoring above average in terms of usability. Although slightly more physically demanding, this method was also judged less frustrating by the users.

However, these input methods were evaluated on natural languages, and thus, linear text. Mathematical expressions, however, are bi-dimensional. Therefore, results from text input literature cannot be directly translated to mathematical expressions. Different approaches to mathematics input already exist. For example, LaTeX is a well-known system able to render complex mathematical expressions [LaT21]. However, LaTeX requires users to learn a specific syntax and have prior knowledge about the structure of mathematical expressions. This overhead increases the cognitive cost of the system [KS98] and is detrimental in our educational context. Additionally, the input is distinct from the final output, which constrains the users to imagine the final output [SF14], and can lead to delayed error detection [KS98].

Windows, Icons, Menus, and Pointers (WIMP) and Drag-and-drop interfaces address this issue by replacing the complex syntax with clickable buttons, and displaying the final output in the user's workspace. Memorizing the position of symbols in potentially large menus, however, still causes cognitive overhead [SF14].

Handwriting-based interfaces bypass these constraints. Handwriting input methods are faster, more accurate, and more enjoyable that keyboard-based methods [AYK05, SNA01, LLM*08]. However, handwriting methods rely on tracking precision. This is no issue for a stylus-based interfaces, but becomes problematic in VR as the input is constrained by the tracking of the hands or the controllers. Some more precise solutions exist [Sen21], but are still expensive and cumbersome.

Finally, Anthony et al. evaluated speech-based input methods for mathematical expressions [AYK05]. However, speech alone can be ambiguous: for example, "x over y + 4" could stand for either $\frac{x}{y+4}$ or $\frac{x}{y}+4$. Moreover, using speech as an input in an educational context is not recommended as it can impair communication with their peers or their teacher.

To our knowledge, the problem on mathematics input in VR has not yet been explored. In this project, we offer a first perspective on this issue, from an educational standpoint.

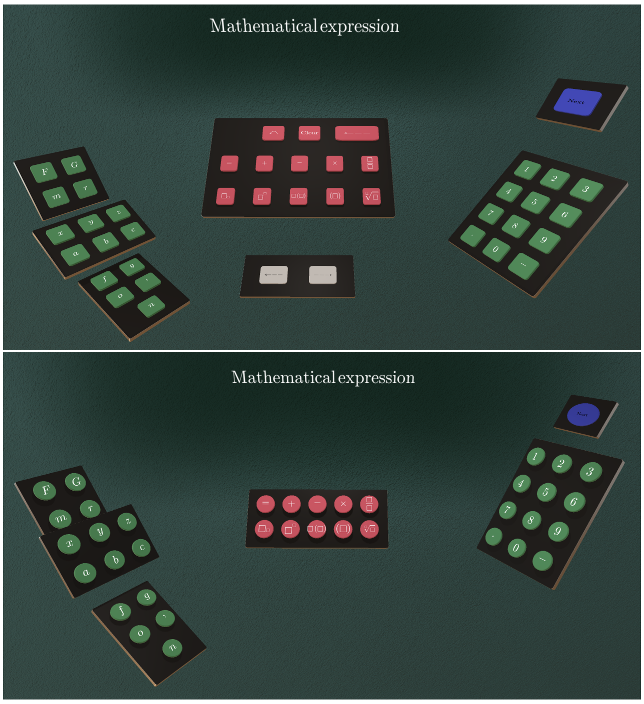


Figure 2: *The keyboard (top) and drag-and-drop (bottom) interfaces*

3. Method and Implementation

For this work, we focus on high-school materials. This means that in order to align with our educational perspective, our interaction techniques follow two core principles:

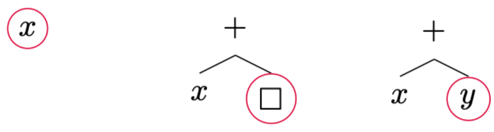
Ease of learning: Users, independently of their experience with mathematics or VR, are able to input complex expressions no later than a few minutes after their first exposition to the system.

Low overhead: Our system does not impair the learning process with an overly cumbersome input method. Entering expressions takes as little time and cognitive load as possible.

3.1. Interaction techniques and Interface

In a first phase, we designed various prototypes for math input in VR, each making usage of proprioception to varying degrees, as recommended by Mine et al. [MBS97]. We focused on controller-based methods as these are available on all VR devices. We selected two prototypes fulfilling our two core principles: a) a Keyboard-like (KBD) interaction, and b) a Drag-and-drop (DND) interaction. We implemented both approaches with the Unity game engine [Uni21] and a Samsung Odyssey+ VR Head Mounted Display (HMD) [Sam21] (Figure 2).

For each technique, we implemented a virtual interface with panels placed around the user, containing the libraries of operators and terms available. The expression is displayed in front of the user. Each element of the expression is surrounded by a semi-transparent bounding box. When the cursor is located on an item, this bounding box is highlighted by changing to a darker shade of gray.

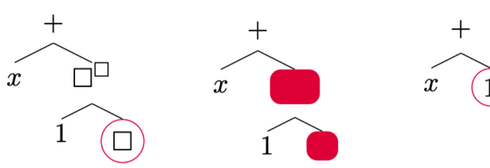


(a) *Initial state* x

(b) *Operator Insertion* $x+\square$

(c) *Term Insertion* $x+y$

Figure 3: *Element insertion. The red circle designates the cursor. The empty squares represent place-holders.*



(a) *Initial state* $x+1^{\square}$

(b) *Operator and parent deletion* $x+1$

(c) *Child replacement* $x+1$

Figure 4: *Element deletion.*

With the KBD interaction technique, inspired by previous research on text-input in VR, the user presses operators and terms on a virtual keyboard using the controllers to add them to the expression. Upon success, a sound is played and haptic feedback is delivered. The user can delete elements by pressing a backspace key, and move the cursor either with the arrow keys, or by directly touching the desired position in the expression.

With the DND interaction technique, inspired by embodied interaction literature, the user presses the grip button on the controller to grab elements, and can then simply move them and place them at their correct position in the expression. Pressing this button involves squeezing the hand as one would to grab an object, thus creating a strong mapping between the natural gesture and the virtual action. In addition, the user can point at items in the expression and press the trigger button to delete them.

3.2. Mathematical expressions

We modeled expressions as trees with operators and terms as nodes.

The terms are represented as a string, and their insertion and deletion are done by simple string manipulation. New operators, however, are inserted in place of the one on which the cursor is positioned, which is then re-positioned as the leftmost child of the new operator (Figure 3). An operator is deleted by removing the empty term to its right and replacing the parent with its leftmost child (Figure 4).

The tree representation of the expression is then used to recursively generate a LaTeX string, rendered using the TEXDraw Unity package [Wel21].

4. Experiment and Results

We explore two hypotheses: both approaches are *easy to learn* ($H1$), KBD follows the *low overhead* principle better than DND ($H2$). In order to avoid learning and fatigue effects, we conducted a quantitative user study using a between-subject design.

4.1. Procedure and Demographics

We recruited 26 unpaid participants and assigned them randomly to each condition: 13 participants used the KBD method, and 13 participants used the DND method.

Before using the system, users filled in a general questionnaire and an Simulator Sickness Questionnaire (SSQ) [KLBL93]. The general questionnaire included demographic questions, a 5-points Likert scale self-assessment of their math knowledge [Lik32], and a question about how often they use VR or movable controllers. The participants profiles are summarized as follows:

- 9 (KBD = 5, DND = 4) participants identified as women, and 17 (KBD = 8, DND = 9) as men.
- 19 (KBD = 10, DND = 9) participants rated their general knowledge of math as 4 or higher, 7 (KBD = 3, DND = 4) as 3 or lower.
- 7 (KBD = 3, DND = 4) participants reported using VR or movable controllers more than once a month, 19 (KBD = 10, DND = 9) as less often.
- KBD participants' ages ranged from 21 to 46 ($M = 28.8y, SD = 7.15y$), whereas DND particpants' ages ranged from 24 to 37 ($M = 29.4y, SD = 4.25y$).

The participants then started the main trial in VR. Using their assigned interaction technique, the participants had to either copy a mathematical expression displayed above the workspace, or change elements of their expression to match a given one. After correctly inputting or correcting the expression, participants moved onto the next one until completion.

We designed the trials to last under 30 minutes in order to avoid fatigue effect. Every user was given the same 23 expressions from high-school mathematics and physics topics. The first ten expressions (Table 2 in appendix) were considered as a warm-up and were not taken into account for our analysis. One expression was dismissed a posteriori due to a measurement error, leaving 12 expressions for our analysis (Table 1). The expressions to copy were mathematically accurate. The expressions to correct were not, but the expected final result was.

In total, the trial lasted around half an hour. Throughout the trial we logged data regarding participants' efficiency (time per task, speed of input, number of superfluous corrections) as well as their general use of the system (dominant hand usage, controller movement, head rotations).

Finally, after the trial, users answered a System Usability Scale (SUS) questionnaire [Bro96], a Raw NASA-TLX (TLX) questionnaire [Har06], and a SSQ questionnaire [KLBL93]. The results of the latter were compared with those of the initial SSQ questionnaire.

4.2. Analysis and Results

We first looked into performance (Tables 3 and 4 in appendix). As all users were able to input the expressions without external help

Task	Type	Requested formula
1	Copy	$x^{\frac{1}{3}} = \sqrt[3]{x}$
2	Copy	$x^6 = o(x_7)$
3	Correction	$x^6 = o\left(x^7\right)$
4	Copy	$(a+b)\times(a-b) = a^2 - b^2$
5	Copy	$y = 2x^2 + 27x - 4$
6	Correction	$f(x) = 2x^2 + 27x - 4$
7	Copy	$x_0 = \frac{-b-\sqrt[2]{b^2-4ac}}{2a}$
8	Copy	$\frac{1}{1-x} = 1 + x + x^2 + ... + x^n + o(x^n)$
9	Copy	$(1+2)\times 3 = 7$
10	Copy	$f(x) = \frac{x}{\sqrt[2]{2}} + 72x^2$
11	Copy	$a^2 + b^2 = c^2$
12	Copy	$F = \frac{G\times m_1 \times m_2}{r^2}$

Table 1: *The list of tasks taken into account for the analysis*

after the warm-up phase ($H1$). With the KBD condition, expression 9 was the quickest to input ($M = 19.28s, SD = 2.80s$), and expression 8 the slowest ($M = 113.95s, SD = 38.33s$). With the DND condition, expression 9 was the quickest ($M = 24.77s, SD = 4.24s$) and expression 8 the slowest ($M = 185.54s, SD = 50.73s$). We then used independent t-tests to compare the time of completion of each task across conditions. We found that users of the KBD condition took significantly less time to input long expressions, namely expressions 4 ($t(26) = -3.159, p = 0.004$), 8 ($t(26) = -3.670, p = 0.001$), and 10 ($t(26) = -3.301, p = 0.003$), as well as the short expression 9 ($t(26) = -3.742, p = 0.001$). This means that, in terms of time, $H2$ is verified.

Looking at the physical involvement, we observed that for every expression but expressions 2 and 3, users of the DND condition made significantly larger hand movements. This is congruent with our expectations ($H2$). However, this difference could be reduced by adapting the distance between the user and their workspace. Regarding motion sickness, there was no significant difference in the SSQ scores across conditions ($p = 0.276$). The KBD method had an average increase of 0.05 in SSQ scores, and the DND 0.12. Such small differences imply that neither methods caused motion sickness, which concurs with the users being mostly static when using the system.

Regarding usability, KBD method scored a mean of 72.31 on the SUS questionnaire and the DND 74.42, which is considered "good" [BKM09]. The difference was not significant ($p = 0.681$), showing that both methods were equally good in terms of usability, despite the efficiency and physical involvement differences.

Moreover, we looked into task load and found no significant difference between the two methods ($p = 0.355$). The average TLX scores were 44.32 for the KBD and 40.48 for the DND, which indicates that the cognitive overhead of the system should be reduced. There can be several reasons for this higher load. First, some participants reported that they chose to remember the expression rather than repeatedly looking at it during the task. This issue is specific to our experiment tasks and should be investigated in a more ecologically valid context. Additionally, participants in the KBD condition reported struggling to perceive the cursor highlighting. In parallel, participants in the DND condition reported alignment issues of the collision boxes. Finally, similarities between operators such as $\square(\square)$ and $(\square)$, or $\square - \square$ and $-\square$, confused some participants.

Despite these flaws, more users reported having fun while using the DND method than with the KBD interface. One DND user even compared placing elements at their correct position to playing a puzzle game. Although anecdotal, these comments suggest that the DND approach should not be discarded solely on the account of lower efficiency.

5. Limitations and Future Work

As our system targets education, it is important that the interaction supports our two core principles, ease of learning and low overhead, even for math and VR novices. The low number of participants among these categories did not allow us to conduct a conclusive analysis, and future work should focus on detecting such biases. We would also like to evaluate our interaction techniques with high-school students, in the context of educational activities in VR and in a more ecologically valid environment as soon as health regulations allow it.

Moreover, we identified a few interface issues such as the placement of the symbol panels, the cursor highlighting, and the misalignment between the bounding boxes and their content. It is possible these issues affected our results. Future work should implement movable panels, stronger visual feedback, and better alignment of the bounding boxes.

Finally, we focused on controller-based interaction as this is the most wide-spread form of interaction in VR. However, other approaches could be considered. For example, hand tracking technologies, either external such as Leap Motion [Ult21], or embedded such as in the Oculus Quest HMD [Ocu21], pave the way for more natural interaction, because, unlike controllers, they do not require an additional abstraction layer.

6. Conclusion

We implemented and evaluated two novel methods for the input mathematical expressions in Virtual Reality, namely a Keyboard-like (KBD) method and a Drag-and-drop (DND) one. A quantitative user study augmented with qualitative feedback from the participants showed that both approaches were usable and did not induce motion sickness. Although the KBD approach was less time-consuming and less-physically demanding, users equally enjoyed the DND approach. Our study also identified a high task load index for both methods and potential reasons responsible for these results.

7. Acknowledgments

The authors would like to thank participants of the user study for their invaluable help, the reviewers for their time and feedback, and the members of the ETH Game Technology Center for participating in our brainstorming sessions.

References

[ANWP*20] ABRAHAMSON D., NATHAN M. J., WILLIAMS-PIERCE C., WALKINGTON C., OTTMAR E. R., SOTO H., ALIBALI M. W.: The future of embodied design for mathematics teaching and learning. In *Frontiers in Education* (2020), vol. 5, Frontiers, p. 147. 1

[AYK05] ANTHONY L., YANG J., KOEDINGER K. R.: Evaluation of multimodal input for entering mathematical equations on the computer. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems* (2005), pp. 1184–1187. 2

[BKM09] BANGOR A., KORTUM P., MILLER J.: Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies 4*, 3 (2009), 114–123. 4

[Bri90] BRICKEN W.: Learning in virtual reality. 1

[Bro96] BROOKE J.: Sus: a "quick and dirty" usability. *Usability evaluation in industry 189* (1996). 3

[DA19] DUBE T. J., ARIF A. S.: Text entry in virtual reality: A comprehensive review of the literature. In *International Conference on Human-Computer Interaction* (2019), Springer, pp. 419–437. 2

[Har06] HART S. G.: Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting* (2006), vol. 50, Sage publications Sage CA: Los Angeles, CA, pp. 904–908. 3

[KGS12] KILTENI K., GROTEN R., SLATER M.: The sense of embodiment in virtual reality. *Presence: Teleoperators and Virtual Environments 21*, 4 (2012), 373–387. 1

[KLBL93] KENNEDY R. S., LANE N. E., BERBAUM K. S., LILIENTHAL M. G.: Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology 3*, 3 (1993), 203–220. 3

[KS98] KAJLER N., SOIFFER N.: A survey of user interfaces for computer algebra systems. *Journal of Symbolic Computation 25*, 2 (1998), 127–159. URL: https://www.sciencedirect.com/science/article/pii/S074771719790170X, doi:https://doi.org/10.1006/jsco.1997.0170. 2

[KSW00] KAUFMANN H., SCHMALSTIEG D., WAGNER M.: Construct3d: a virtual reality application for mathematics and geometry education. *Education and information technologies 5*, 4 (2000), 263–276. 1

[LaT21] LaTeX: The latex project. https://www.latex-project.org, Retrieved 26 April 2021. 2

[Lik32] LIKERT R.: A technique for the measurement of attitudes. *Archives of psychology* (1932). 3

[LLM*08] LABAHN G., LANK E., MACLEAN S., MARZOUK M., TAUSKY D.: Mathbrush: A system for doing math on pen-based devices. In *2008 The Eighth IAPR International Workshop on Document Analysis Systems* (2008), IEEE, pp. 599–606. 2

[MBS97] MINE M. R., BROOKS F. P., SEQUIN C. H.: Moving objects in space: Exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (USA, 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., p. 19–26. URL: https://doi.org/10.1145/258734.258747, doi:10.1145/258734.258747. 2

[MP14] MALINVERNI L., PARES N.: Learning of abstract concepts through full-body interaction: A systematic review. *Journal of Educational Technology & Society 17*, 4 (2014), 100–116. 1

[Ocu21] OCULUS: Hand tracking in unity. https://developer.oculus.com/documentation/unity/unity-handtracking/, Retrieved 28 May 2021. 4

[Rot01] ROTH W.-M.: Gestures: Their role in teaching and learning. *Review of Educational Research 71*, 3 (2001), 365–392. URL: https://doi.org/10.3102/00346543071003365, arXiv:https://doi.org/10.3102/00346543071003365, doi:10.3102/00346543071003365. 1

[Sam21] SAMSUNG: Odyssey+ head mounted display. https://www.samsung.com/us/support/computing/hmd/hmd-odyssey/hmd-odyssey-plus-mixed-reality/, Retrieved 28 May 2021. 2

[Sen21] SENSORYX: Vrfree. https://www.sensoryx.com/finger-tracking/, Retrieved 26 April 2021. 2

[SF14] SHIRAI S., FUKUI T.: Development and evaluation of a web-based drill system to master basic math formulae using a new interactive math input method. In *Mathematical Software – ICMS 2014* (Berlin, Heidelberg, 2014), Hong H., Yap C., (Eds.), Springer Berlin Heidelberg, pp. 621–628. 2

[SFZK18] SPEICHER M., FEIT A. M., ZIEGLER P., KRÜGER A.: Selection-based text entry in virtual reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018), pp. 1–13. 2

[SNA01] SMITHIES S., NOVINS K., ARVO J.: Equation entry and editing via handwriting and gesture recognition. *Behaviour & information technology 20*, 1 (2001), 53–67. 2

[Spi21] SPIEL K.: The bodies of tei–investigating norms and assumptions in the design of embodied interaction. In *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction* (2021), pp. 1–19. 1

[SSE18] STERNIG C., SPITZER M., EBNER M.: Learning in a virtual environment: implementation and evaluation of a vr math-game. In *Virtual and Augmented Reality: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2018, pp. 1288–1312. 1

[TSB17] TRAN C., SMITH B., BUSCHKUEHL M.: Support of mathematical thinking through embodied cognition: Nondigital and digital approaches. *Cognitive Research: Principles and Implications 2*, 1 (2017), 1–18. 1

[Ult21] ULTRALEAP: Leap motion controller. https://www.ultraleap.com/product/leap-motion-controller/, Retrieved 28 May 2021. 4

[Uni21] UNITY: Unity game engine. https://unity.com/, Retrieved 28 May 2021. 2

[Wel21] WELLOSOFT: Texdraw. https://assetstore.unity.com/packages/tools/gui/texdraw-51426, Retrieved 26 April 2021. 3

8. Appendix

Type	Requested formula	Type	Requested formula
Copy	$x+y$	Copy	$(x-1)\times 2$
Copy	$\left(\frac{1}{2}\right)\times z$	Copy	$\left(\frac{1}{2}\right)=0.5$
Copy	$a_0=a_1=a_2$	Correction	$\frac{1}{2}=0.5$
Copy	$x+y=y-x$	Copy	$\sqrt[5]{x^5}=x$
Correction	$x+y=y+x$	Copy	$\left(x^2\right)=2x$

Table 2: *The list of tasks used as a warm-up for the participants*

Task	Keyboard	Drag and drop	Statistic	p-value
1	M = 41.77, SD = 15.83	M = 50.43, SD = 18.23	$t(26)=-1.242$	$p=0.226$
2	M = 91.60, SD = 52.82	M = 90.05, SD = 51.19	$t(26)=0.073$	$p=0.942$
3	M = 18.15, SD = 18.91	M = 26.49, SD = 17.05	$t(26)=-1.135$	$p=0.268$
4	**M = 51.99, SD = 15.86**	M = 88.28, SD = 36.50	$t(26)=-3.159$	$p=0.004$
5	M = 54.23, SD = 33.45	M = 70.38, SD = 32.89	$t(26)=-1.192$	$p=0.245$
6	M = 11.01, SD = 5.30	M = 22.04, SD = 22.46	$t(26)=-1.656$	$p=0.111$
7	M = 155.02, SD = 69.13	M = 167.33, SD = 71.42	$t(26)=-0.429$	$p=0.672$
8	**M = 113.95, SD = 38.33**	M = 185.54, SD = 50.73	$t(26)=-3.670$	$p=0.001$
9	**M = 19.28, SD = 2.80**	M = 24.77, SD = 4.24	$t(26)=-3.742$	$p=0.001$
10	**M = 52.42, SD = 9.53**	M = 87.95, SD = 36.04	$t(26)=-3.301$	$p=0.003$
11	M = 28.12, SD = 11.57	M = 33.90, SD = 9.51	$t(26)=-1.336$	$p=0.194$
12	M = 50.79, SD = 28.97	M = 67.14, SD = 24.78	$t(26)=-1.486$	$p=0.150$

Table 3: *Comparison across conditions of the time taken to complete a task, in seconds. The better results are written in bold when significant.*

Task	Keyboard	Drag and drop	Statistic	p-value
1	**M = 0.64, SD = 0.13**	M = 0.92, SD = 0.14	$t(26)=-4.930$	$p<0.001$
2	M = 1.12, SD = 0.47	M = 1.39, SD = 0.58	$t(26)=-1.204$	$p=0.240$
3	M = 0.37, SD = 0.18	M = 0.51, SD = 0.22	$t(26)=-1.841$	$p=0.078$
4	**M = 0.81, SD = 0.20**	M = 1.61, SD = 0.46	$t(26)=-5.513$	$p<0.001$
5	**M = 0.73, SD = 0.25**	M = 1.29, SD = 0.39	$t(26)=-4.185$	$p<0.001$
6	**M = 0.38, SD = 0.09**	M = 0.61, SD = 0.32	$t(26)=-2.403$	$p=0.024$
7	**M = 1.81, SD = 0.70**	M = 2.59, SD = 0.90	$t(26)=-2.367$	$p=0.026$
8	**M = 1.51, SD = 0.53**	M = 3.15, SD = 0.67	$t(26)=-6.619$	$p<0.001$
9	**M = 0.44, SD = 0.07**	M = 0.70, SD = 0.10	$t(26)=-7.550$	$p<0.001$
10	**M = 0.81, SD = 0.09**	M = 1.63, SD = 0.50	$t(26)=-5.506$	$p<0.001$
11	**M = 0.58, SD = 0.12**	M = 0.88, SD = 0.15	$t(26)=-5.564$	$p<0.001$
12	**M = 0.81, SD = 0.26**	M = 1.38, SD = 0.30	$t(26)=-4.993$	$p<0.001$

Table 4: *Comparison across conditions of the controller movement distance per task, in meters. The better results are written in bold when significant.*