17th-19th June 2024
Gliwice-Szczyrk, Poland
DEPARTMENT OF ENGINEERING MATERIALS AND BIOMATERIALS
FACULTY OF MECHANICAL ENGINEERING
SILESIAN UNIVERSITY OF TECHNOLOGY
INTERNATIONAL STUDENTS SCIENTIFIC CONFERENCE

Optimization of numeric keyboard interfaces based on signals from eye tracking components built in virtual reality goggles

Maja Rudnicka[a], Michał Lasak[b], Szymon Sadza[c], Dariusz Myszor[d]

[a]Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science,, SKN SpaceCoffee
email: maja.rudnicka@alogliwice.polsl.pl
[b]Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science,, SKN SpaceCoffee
email: michal.lasak@alogliwice.polsl.pl
[c]Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science, SKN SpaceCoffee
email: szymon.sadza@alogliwice.polsl.pl
[d]Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science, Department of Algorithmics and Software
email: dariusz.myszor@polsl.pl

Abstract: This article presents results of tests of various numeric keyboard interfaces that were created in the Unity Engine. Those keyboards were navigated through utilizing virtual reality goggles with eye tracking sensors built into them.

Keywords: VR, Eye Tracking, eye activity

1. INTRODUCTION

This paper delves into the development of different types of numeric keyboard interfaces designed for entering numbers utilizing Eye Tracking technologies, all crafted within the Unity engine. Eye tracking[1] is a non-invasive method of determining the direction of a user's gaze by detecting the reflection of infrared light off of the user's eye as pupils absorb a fixed amount of light which makes them invaluable as a point of reference. Eye tracking technologies by themselves have been around for a moment, but it was only recently that they became integrated with Virtual Reality goggles[2]. That mend of both provides researchers with new possibilities in terms of alternative ways of computer inputting.

2. DESIGNS

The Designs were prepared in the Unity Engine, seven different designs were created, with the limitation of 10 numbers per numeric keyboard, so that every number is of comfortable dimensions. Looking at a specific button allows the detector to determine the position of the pupil in relation to button location. That allows the user to determine which button is chosen by the user.

2.1 First design

It's a standard number keyboard design, but it lacks a "0", which makes the count of number combinations smaller. It's possible to implement a way to write zeros, for example if pressing "enter" would pass to the next number and zero is the default.

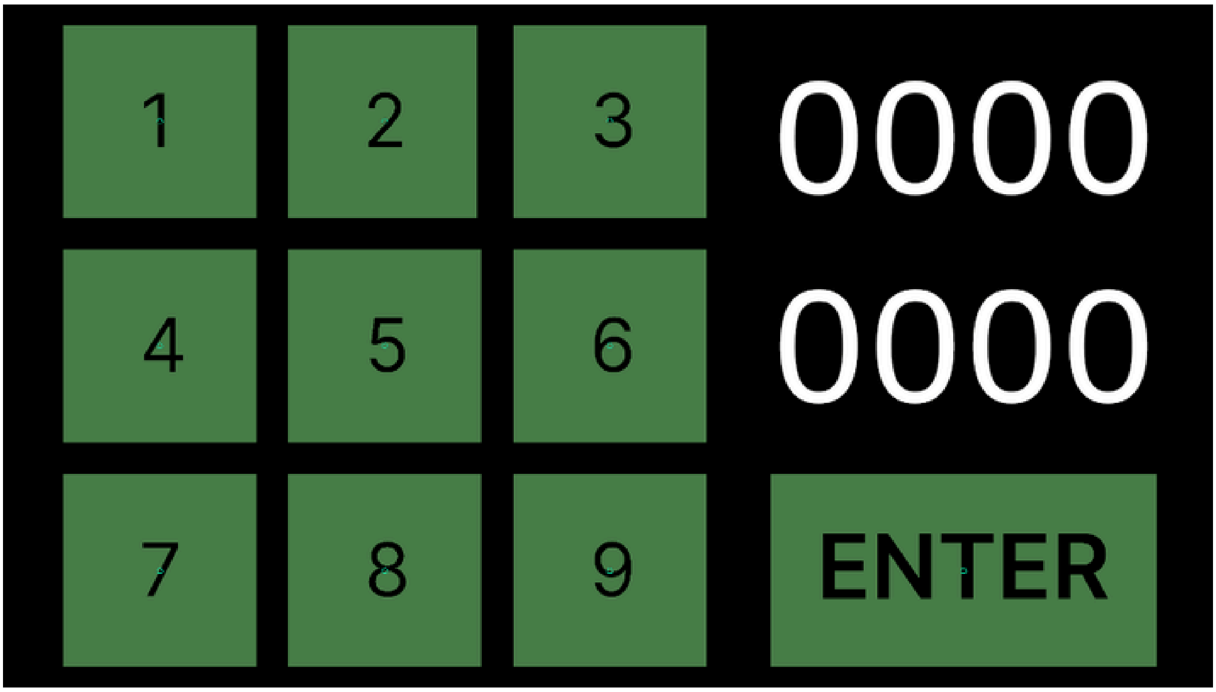

Figure 1. First Design of a BCI numerical keyboard

2.2 Second design

Second design allows to write much more precisely and complexly, but there is a possibility of it being slower than other designs, however it allows to implement in the future more buttons, since in none of the menus the limit of ten buttons is used. This design has one main menu [Fig. 2] and several sub menus [Fig. 3-4].

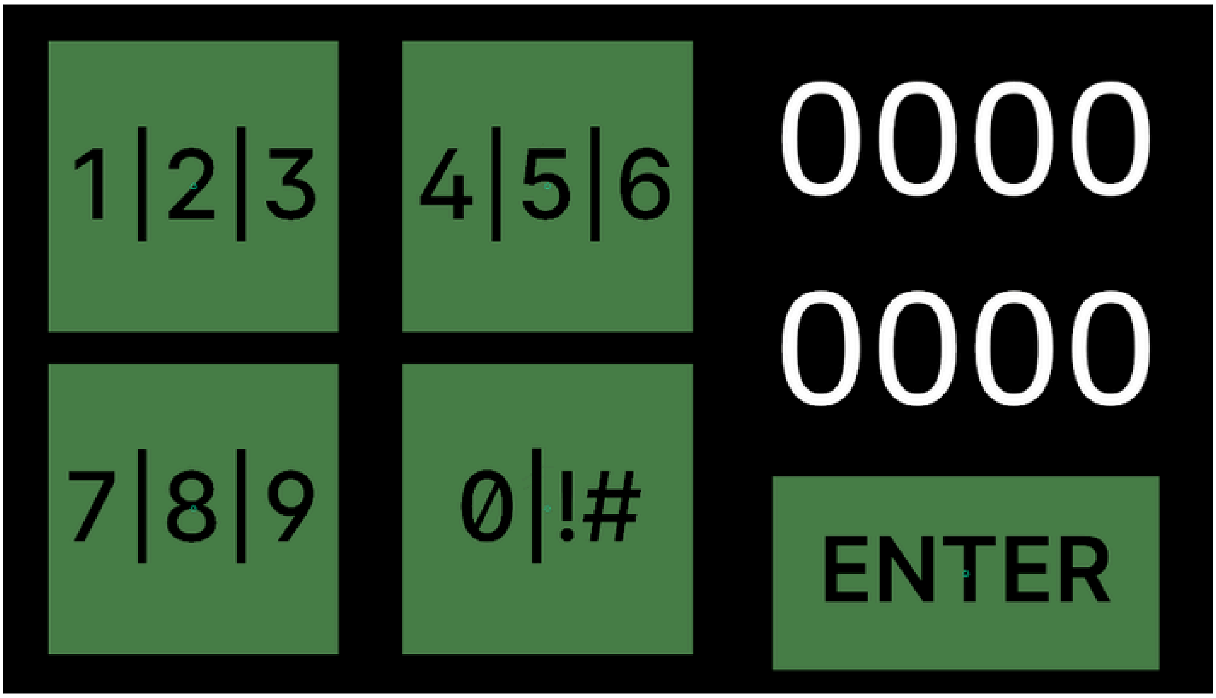

Figure 2. Second Design of a BCI numerical keyboard, in the main menu

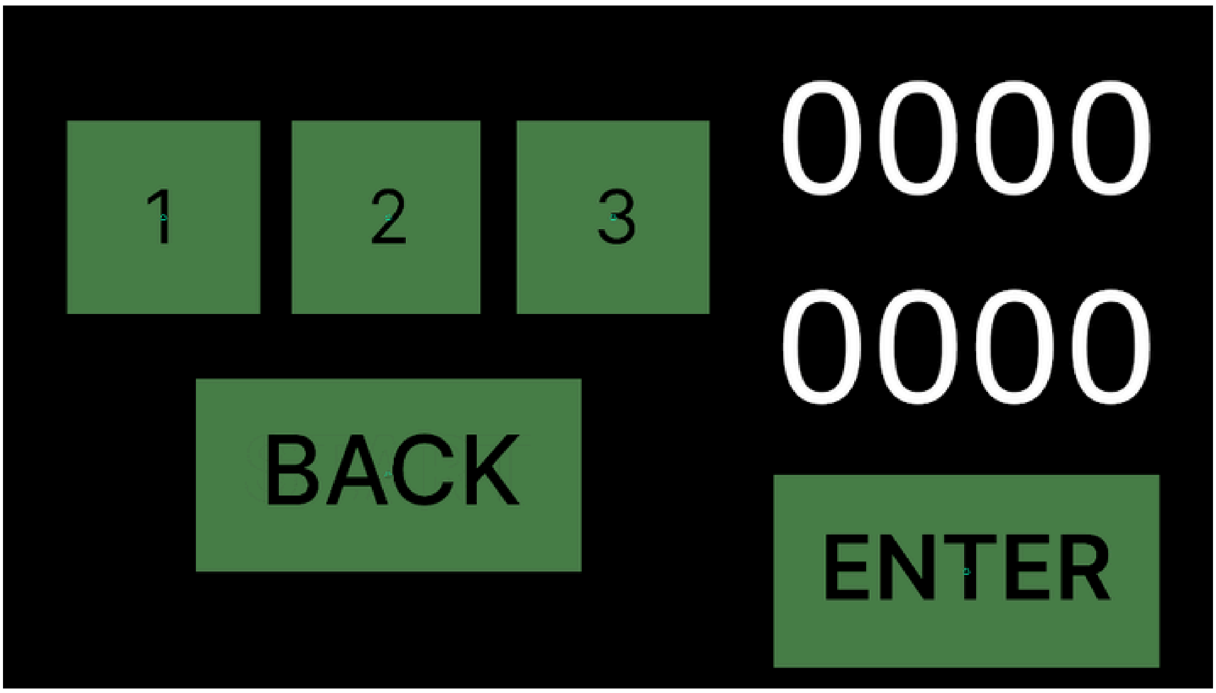

Figure 3. Second Design of a BCI numerical keyboard, in the first menu

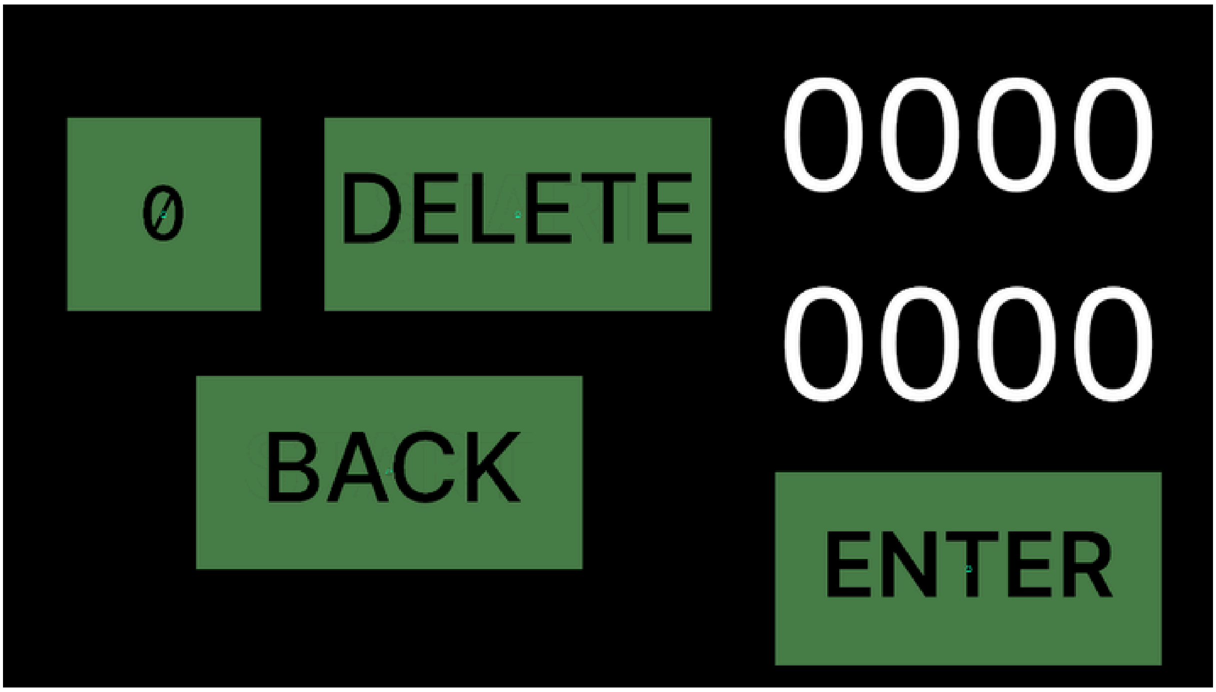

Figure 4. Second Design of a BCI numerical keyboard, in the last menu

2.3 Third design

The Third design is a blend between the first two, it uses similar menus as the Second Design, but it also has a typical layout that people are used to. It lacks the possibilities of the Second Design, but it can be advantages to those that are used to such.

Figure 5. Third Design of a BCI numerical keyboard, with the first menu open

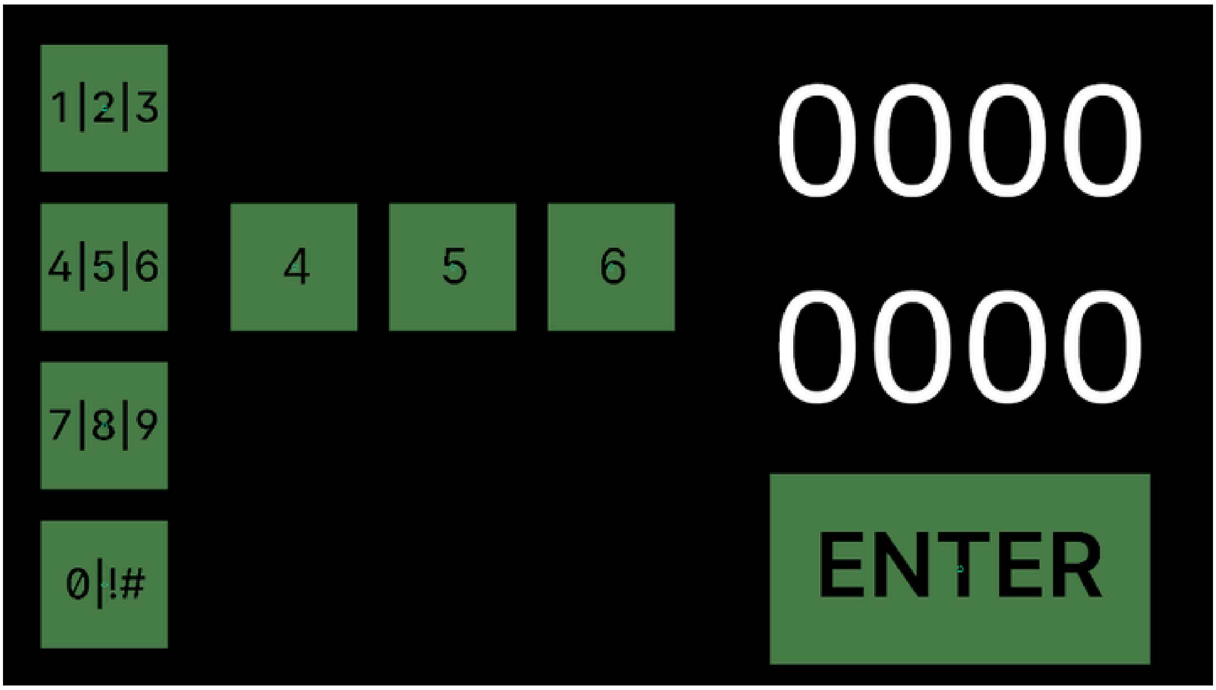

Figure 6. Third Design of a BCI numerical keyboard, with the second menu open

2.4 Fourth design

The Fourth Design is a simplified version of the Second, with only two menus and a simpler design. It still allows for some future improvement, but it is less flexible. It has a main menu [Fig.7] and two sub menus [Fig. 8]

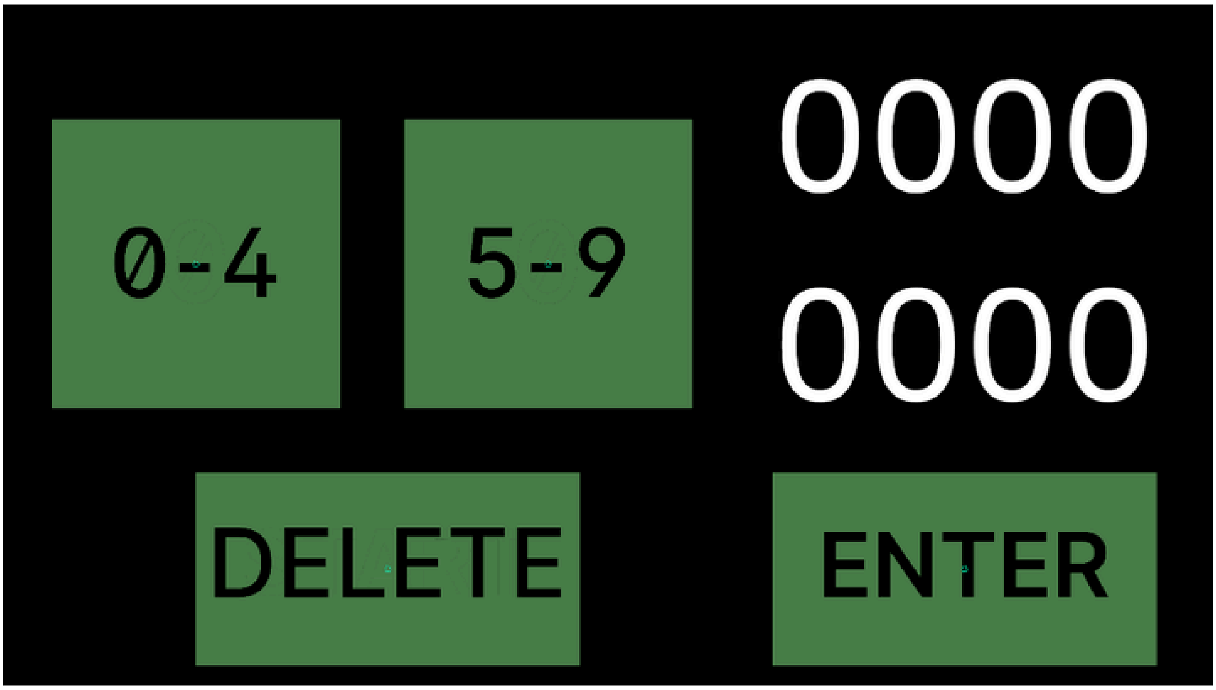

Figure 7. Fourth Design of a BCI numerical keyboard, with the main menu open

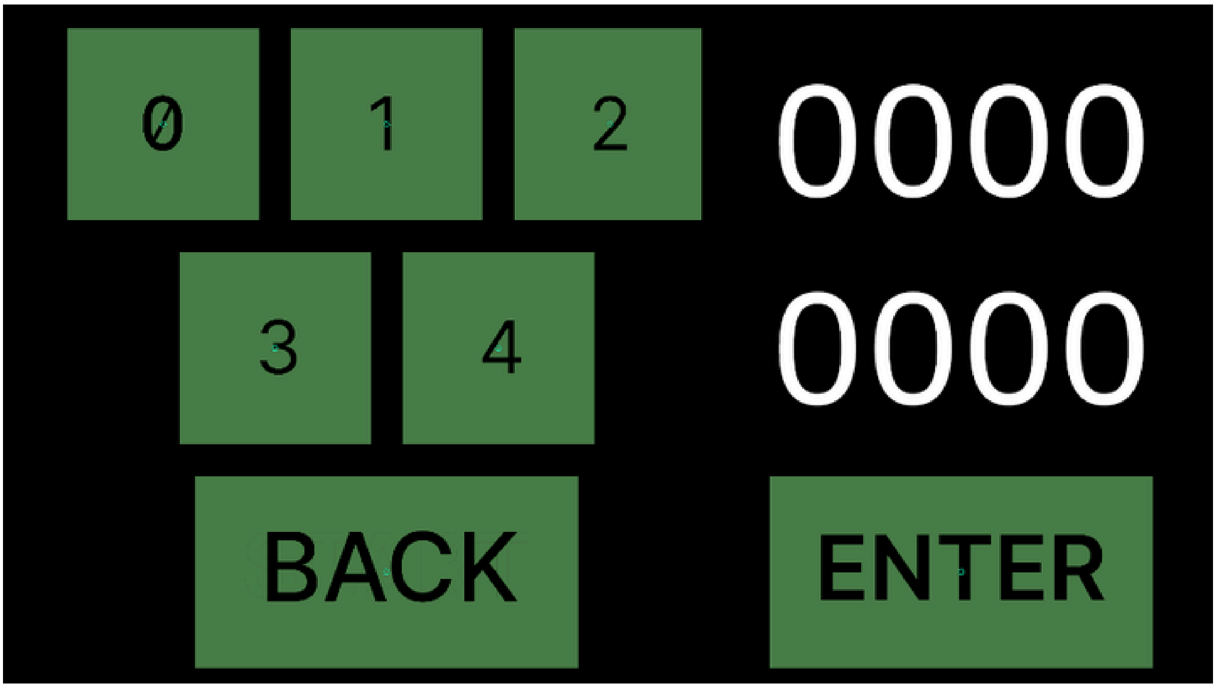

Figure 8. Fourth Design of a BCI numerical keyboard, with the first sub menu open

2.5 Fifth design

The Fifth Design is based on the First one, but with a zero instead of an enter, which provides another problem with the number selection, because every number can be selected only once.

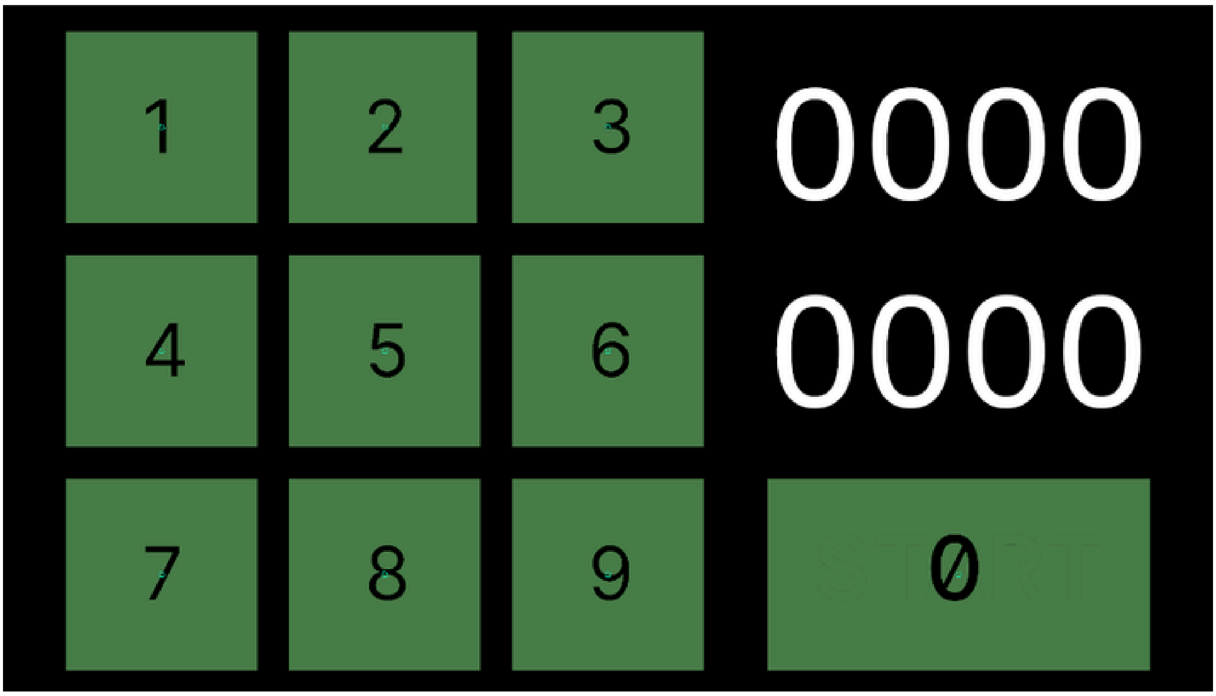

Figure 9. Fifth Design of a BCI numerical keyboard

2.6 Sixth design

The Sixth Design is a different version of the Fourth, with two menus in a circle. It is more intuitive, but allows for even less flexibility with additional buttons. It's reaching the 10 buttons limit.

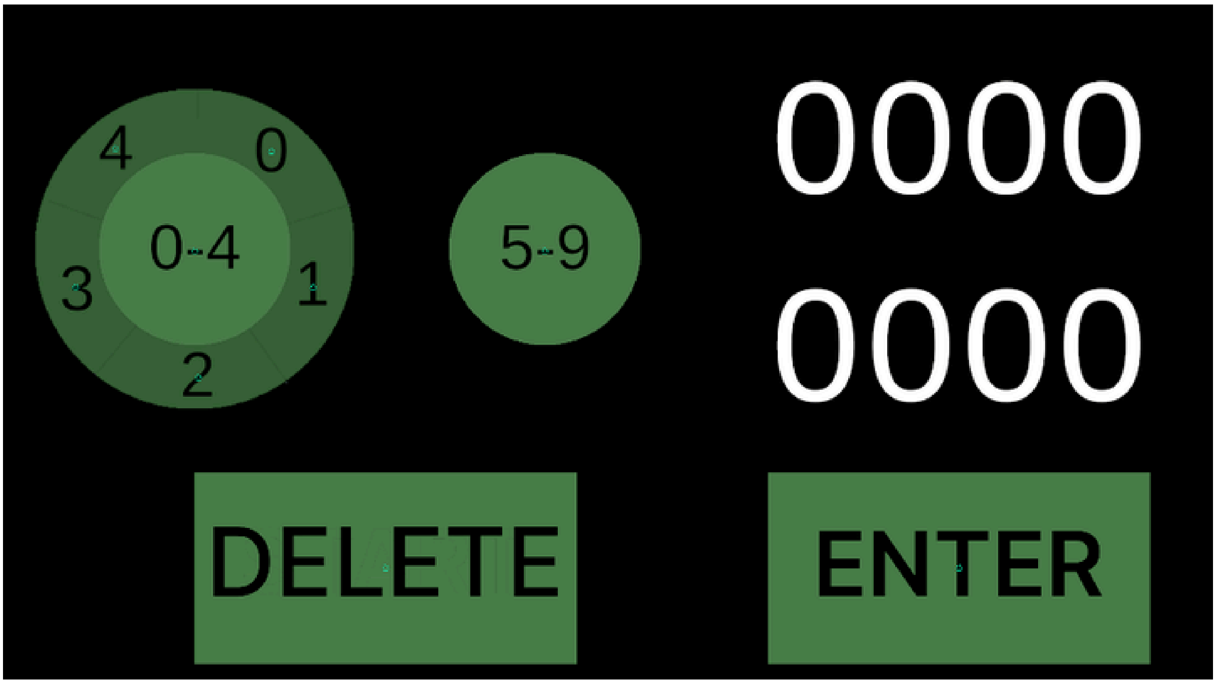

Figure 10. Sixth Design of a BCI numerical keyboard, with the first part expended

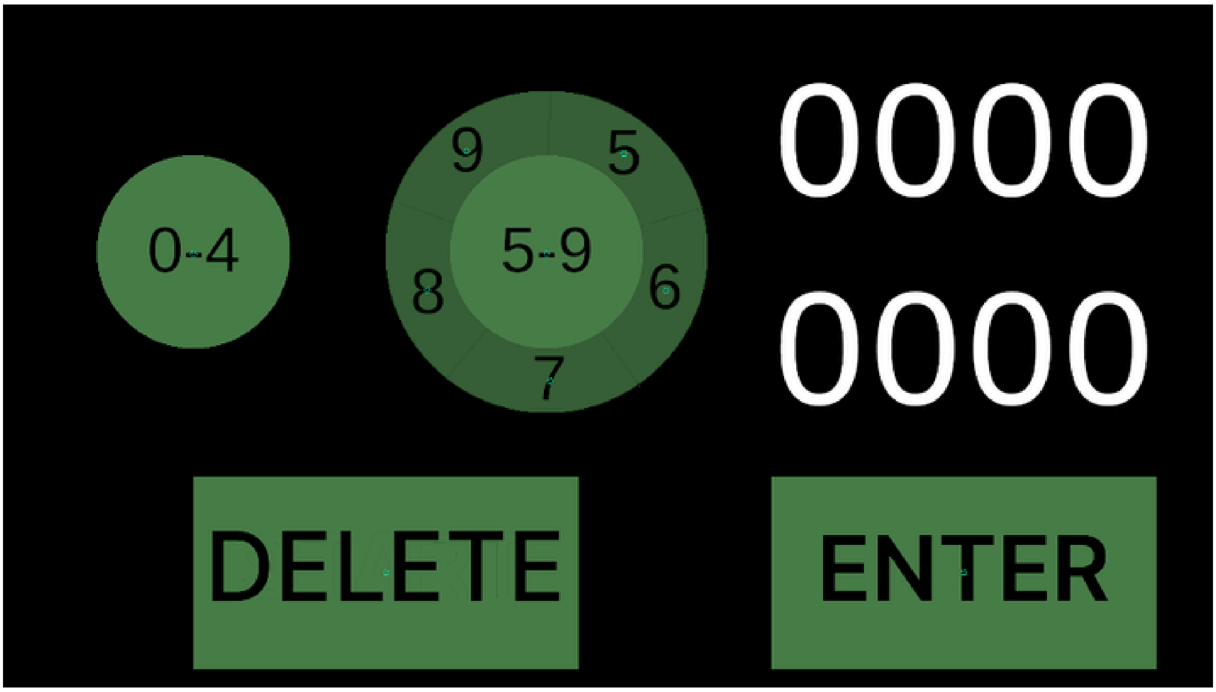

Figure 11. Sixth Design of a BCI numerical keyboard, with the second part expended

2.7 Seventh design

The Seventh Design is similar to the Fifth Design, it has a different layout, that could be more effective and intuitive.

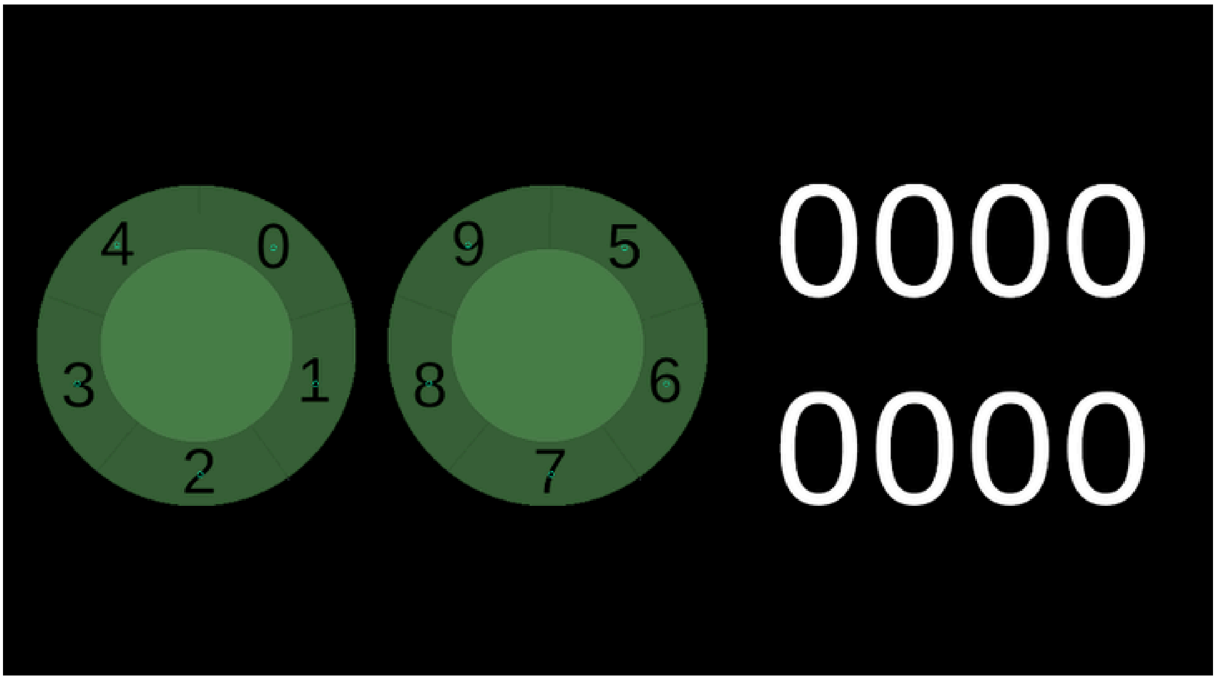

Figure 12. Seventh Design of a BCI numerical keyboard

3. FUTURE FURTHER DEVELOPMENTS

The next step of development is to test all of the designs and choose the best ones to develop them further. Few possible further developments include extending the program to include more combinations and buttons, that's especially likely, if the keyboards based on menus and submenus prove effective or if new designs become comfortable to use. Another way of development is a more personalized interface, customizable by the user, that would allow for the best effectiveness, but would require better calibration and would be less objective in tests.

4. CONCLUSION

Various designs of Eye Tracking numeric keyboard interfaces were developed within the Unity Engine, Each design offers unique features and trade-offs, ranging from simplicity to flexibility. The future of this technology lies in rigorous testing to identify the most effective designs for further refinement. These developments hold promise for improving accessibility and communication for individuals with disabilities, marking a significant step forward in Eye Tracking technology.

BIBLIOGRAPHY

1. Duchowski, A. T. (2017). *Eye Tracking Methodology: Theory and Practice* (3rd ed.). Springer. https://doi.org/10.1007/978-3-319-57883-5

2. VR headsets such as the HTC Vive Pro Eye or Meta Quest Pro now incorporate built-in eye-tracking sensors, enabling immersive gaze-controlled interfaces.📖 *Example Study:* Pfeuffer et al. (2015), *Gaze+ Pinch Interaction in VR*.

3. Majaranta, P., & Räihä, K.-J. (2007). Text entry by gaze: Utilizing eye-tracking. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, 2007, 1165–1172. https://doi.org/10.1145/1240624.1240807

4. Zhang, Y., Skadron, K., & Huang, T. S. (2020). Gaze typing: A comparative evaluation of eye-tracking input methods. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 27(4), Article 26. https://doi.org/10.1145/3399706

5. LoPresti, E. F., Brienza, D. M., Angelo, J., & Gilbertson, L. (2008). Interface design for eye gaze input. *Assistive Technology*, 20(1), 12–22. https://doi.org/10.1080/10400435.2008.10131927

6. Portillo-Lara, R., & González, P. A. (2019). Unity for Human-Computer Interaction prototyping: A survey. *Journal of Computer Languages*, 52, 20–33. https://doi.org/10.1016/j.cola.2019.100711

7. Majaranta, P., & Bulling, A. (2014). Eye tracking and eye-based human–computer interaction. In *Advances in Physiological Computing* (pp. 39–65). Springer. https://doi.org/10.1007/978-1-4471-6392-3_3