IIOO IS NOT HERE - Building and exhibiting media art installation with novel modular tangible programming interface

Building and exhibiting media art installation with novel modular tangible programming interface

Tuomo K. A. Rainio*
Lecturer, Academy of Fine Arts, University of the Arts Helsinki, tuomo.rainio@uniarts.fi

Tero T. Niskanen**
Technician, Academy of Fine Arts, University of the Arts Helsinki, tero.niskanen@uniarts.fi

Abstract

The installation IIOO IS NOT HERE deals human-machine relations and interfaces. It takes a critical standpoint to subject-object relation and questions of control and elaborates N. Katherine Hayles' concept of cognitive assemblage in the context

* Tuomo Rainio (b. 1983, Espoo) is a visual artist based in Helsinki. His art works combine lens-based media with programming. He works with translations between concepts, codes and images. The art works study the conditions of time, space and perception. Rainio is interested in the concept of noise in the visual realm. His art works present subjects in the constant state of becoming. The images are in the state of transformation where they are not-yet-representations or not-anymore-representations.

** Tero Niskanen (b. 1982, Tampere) is Helsinki based artist and experimental musician. He graduated from the Academy of Fine Arts Helsinki in 2018. Main themes in Niskanen's artistic practise are perception, interaction and materiality. Typical for his works are processes consisting of series of experiments, cross pollination of mediums and the situations of intertwining.

of media art installation. The installation is built with newly developed tangible programming interface called Pino. During the exhibition artist will reconfigure the installation and change its behavior. The performative act of artist re-configuring the installation live will highlight the use and possibilities of modular tangible user interface in the context of interactive art installation.

CCS CONCEPTS

•Applied computing~Arts and humanities~Media arts • Human-centered computing~Interaction design~Empirical studies in interaction design • Human-centered computing~Collaborative and social computing~Collaborative and social computing devices

Additional Keywords and Phrases:

interactive installation 1, tangible programming 2, tangible user interface 3, posthumanism 4, modularity 5, physical computing 6

1 Introduction

The installation *IIOO IS NOT HERE* deals human-machine relations and interfaces, it takes a critical standpoint to subject-object relation and questions of control.

IIOO IS NOT HERE is an interactive installation that operates in multiple different configurations that artists have named *awake, idle, sleepy, dream, done*. The configurations reflect both computational terms and human states of mind. In practice they introduce different level of interaction. When awake the installation observes its surroundings with multiple input sensors and can interact with the exhibition visitors. In this state the installation seems responsive and easy to approach. During the dream installation acts as if it would be dreaming, the installation starts to hallucinate and function unexpected ways. It becomes introvert and hard to grasp. Other configurations are variations between the awake and dream states. The transition from one state to another is realized by performative act where artist present will reorganize the part of the installation according to planned actions. This ability for the installation to take different configurations is based on the modularity of the technical system used in the installation (the technical arrangements are described more in detail below). This conceptual division to different states of mind suggests that the human-machine relations are more complex than they first might appear. How does a machine appear when it follows dreamlike logic?

Figure 1: Detail of the installation *IIOO IS NOT HERE*

2 2. BACKGROUND

The installation takes it inspiration from Katherine N. Hayles' concept *nonconscious cognitive assemblage*. The installation does not have centralized control unit, but it consists of multiple smaller units and relays on their interaction. Further its functionality relies not only on the predefined algorithm but rather constantly interacts and evolves with its environment both the exhibition space and online data resources.

Continuing the definition of posthumanism, Hayles started in the book ***How we became posthuman***, Hayles introduces in her more recent book ***Unthought*** the key concept of *cognitive assembles.* The key idea of cognitive assembles is to put highlight on entities formed by systems and individual actors, which may include human and non-human cognitive actors. Challenging an anthropocentric perspective, Hayles' expanded understanding of nonconscious *opens it to comparisons with other biological cognizers on the one hand and on the other to the cognitive capabilities of technical systems* [1]. Hayles underlines the intermediate space among cognitions and their multiple strategies to intertwine as an alternative to the anthropocentric humanist understanding of cognition as thinking or intelligence. Hayles divides between conscious and nonconscious cognitions to emphasize the cognitions as merely information processor. She does not try to argue for similarities between conscious and nonconscious cognizers but rather looks for structural and functional likeness [2].

We find Hayles' notion inspiring in the context of media art, as it makes the human machine relation more nuanced and intertwined. It introduces new perspective to critically reflect what does it mean to be a human among the machines. It also puts emphasis on the technical infrastructures that nowadays we are more and more depend on.

Figure 2: Illustration of the installation *IIOO IS NOT HERE*

3 TECHNICAL REALIZATION OF INSTALLATION

The technical realization of the installation *IIOO IS NOT HERE* further explores the topics discussed above and is profoundly intertwined with the themes of the artwork. The installation is built with new modular system developed at the University of the Arts Helsinki by the authors. The Pino (a Finnish word for "stack") system is designed for artist by artist and uses tangible programming interface for building computational logic and data based interactive installations. Currently the project is in beta state and used for inhouse projects only. The installation *IIOO IS NOT HERE* is the first occasion Pino modules will be used in a public exhibition.

3.1 Background of technical setup

The Pino started as pedagogical project in 2022 and it was designed for post-digital and post-pandemic world, where disassociating the digital from the screen we consider as one of the key pedagogical challenges of today. Pino bridges the gap between physical interaction and digital logic of computation. The system consists of physical building blocks that allow user to build complex systems by stacking modules together. The project in follows the footsteps of the historical lines of tangible programming interfaces (TUIs) from Logo to Cricket and Tangible Programming Bricks [3][4].

The physical computing platforms, open-source hardware design and creative coding frameworks, developed in the past two decades[*], have provided a low threshold learning environments and promoting user-friendly communities for beginners. Nevertheless, by our experience they still introduce a steep learning-curve, when moving away from digital environment and realizing physical computing projects or interactive installations. Democratization of the production of media installations in mind, the Pino modules were designed robust and scalable to fit the demands of professional use

[*] Eg. VVVV, 1998, Processing 2001, Arduino 2005, Openframeworks 2005, TouchDesigner 2008

but without compromising the learning experience. The Pino modules create behavior when connected. The functionality is defined by the order of module arrangement in the stack, the setting of parameters in modules and the external connections between modules. The design supports the idea of learning path from physical interface to coding and building DIY electronics. We see the Pino system not only as a multipurpose toolbox but rather as a learning platform for creators to test their ideas with physical building blocks. Conceptually the Pino system shares a similar approach with previous research on TUIs such as the System Blocks by Lifelong Kindergarten at MIT Media Laboratory and the Tower System developed at the Grassroots Invention group at the MIT Media Lab[6][7].

Various open-source ecosystems are built around low power electronics and microcontrollers[†]. Still little effort has been made to integrate these systems together and, in the contrary, they seem to be marketed as ecosystems of their own. One of the key ideas behind the Pino is to introduce processor and hardware independent ecosystem where users are encouraged to build their own module and make it function with other pre-existing ones. The focus of the Pino modules is not on individual module but the assemblage they create when connected.

The Pino modules are connected by stacking them. The stack is controlled by logic signal that runs across the stack from top to bottom. Logic signal, indicated with light inside module, is used to trigger activity or control the functionality. Some modules are used as logic gates to build more complex systems. In addition to the logic signal some modules can be connected via ethernet cable to transfer data. Data signal can be generated algorithmically, or it can be based on sensor. Data can be also open data acquired from online sources and it can be arithmetically transformed or used to control outputs such as light intensity, motor speed. Logic signal is based solely on transistors whereas data signal uses serial UART protocol that is compatible across multiple platforms. Data signal can be also converted to logic signal with analog to digital converter.

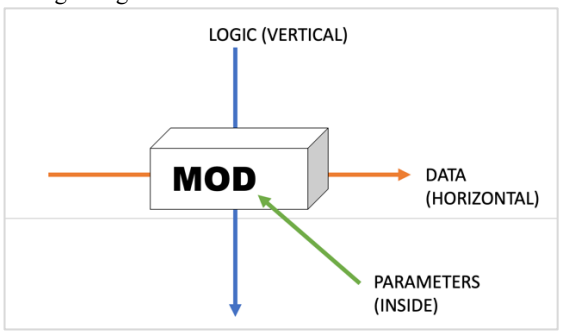

Figure 3. The conceptual structure of signal transfer and manipulation in the Pino system.

One stack of Pino can be easily conceptualized by having three dimensions (see Fig.3). Vertical dimension is reserved for logic signal. Horizontal axis represents data flow. And the third axis represent the parameters in the module. This design principle is used in all different modules. On some occasions one linear program built with one stack is not enough for realizing a complex system. To solve that, the Pino stack can be extended to multiple stacks and both logic and data signal can be shared across multiple stacks. This makes it possible to run parallel processes or built functions that are called by main stack and return value after finishing. This opens interesting pedagogical possibilities to demonstrate the programming conventions in a tangible form.

Tangible programming environments often focus on classroom learning cases and targeted to children or student in age 7-18 years. In most cases their focus is on computational logic, leaving out the possibility to integrate rich audio-visual media content to the system[8][9][10]. Pino tries both to scale out from classroom use to professional context and it also introduce integration to different multimedia systems and industry standard protocols.

[†] Eg. Arduino 2005, Raspberry Pi 2006, Teensy 2008

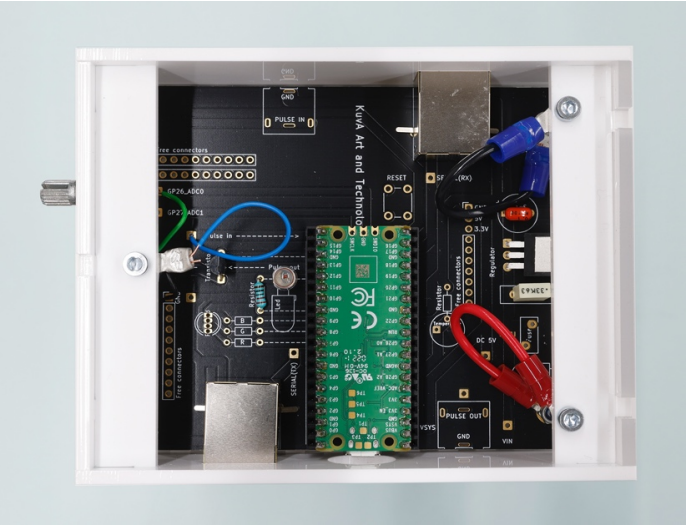

Figure 4: Detail of the Pino module from the top without lid. This configuration is equipped with custom made shild for Raspberry Pico microcontroller.

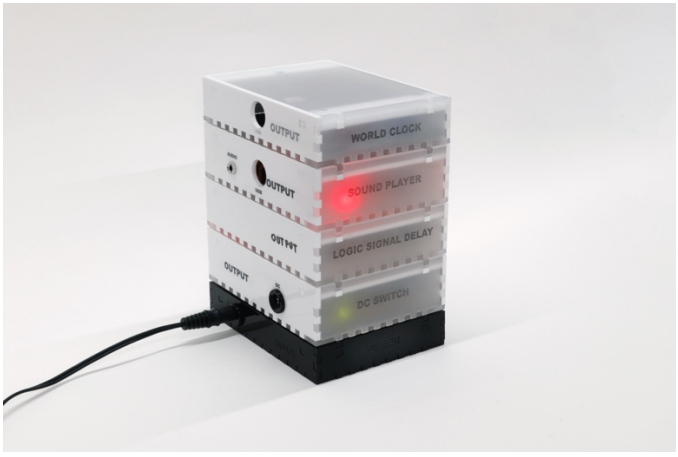

Figure 5: Stack of Pino modules with power module at the base. The stack in the example uses the clock module on the top to trigger logic signal once in hour and further start the playback of sound file in the next module. As the sound starts, it triggers delay timer and after the delay is finished, logic signal further sets the switch module on.

4 SUMMARY

To reflect the current development in the screen-oriented digital technology we try to suggest counter strategies that bring new interfaces for more democratic media art installation. The installation *IIOO IS NOT HERE* reflects the post-humanist concept of cognitive assemblage both on the level of the abstract content of the artwork and on the level of technical realization. The performative act of artists reconfiguring the installation live will highlight the use and possibilities of modular tangible user interface in the context of interactive art installation.

ACKNOWLEDGMENTS

We would like to thank Roberto Pugliese Fusco, Alex Schellong, Joakim Pusenius, Aleksi Pihkanen, Sofia Haapamäki, and Laura Dahlberg for sharing their knowledge, time and expertise to support the project.
This research could not have been done without the financial support University of the Arts Helsinki.

REFERENCES

Hayles, N. Katherine. 2017. Unthought the power of the cognitive nonconscious, The University of Chicago Press, Chicago, 11

Hayles, N. Katherine. 2017. Unthought the power of the cognitive nonconscious, The University of Chicago Press, Chicago, 13-14

McNerney, Timothy Scott. 2004. From turtles to Tangible Programming Bricks: explorations in physical language design. Personal Ubiquitous Computing, 8(5), Springer-Verlag (2004), pp 326–337.

McNerney, Timothy Scott. 2000. Tangible Programming Bricks: an approach to making programming accessible to everyone. Masters thesis, MIT, Cambridge, Massachusetts. Available at http://www.media.mit.edu/people/mc/tangible-programming. html

Zuckerman, Oren and Resnick, Mitchel. 2003. System Blocks: A Physical Interface for System Dynamics Learning, MIT Media Laboratory. Available at https://dam-prod2.media.mit.edu/x/files/papers/Zuckerman_SystemBlocks_ISDC03.pdf

Mikhak, B., Lyon, C., Gorton, T. (2002). The Tower System: a Toolkit for Prototyping Tangible User Interfaces. Submitted as a long paper to CHI 2003. Also see http://gig.media.mit.edu/projects/tower/index.html

Zuckerman, Oren and Resnick, Mitchel. 2003. System Blocks: A Physical Interface for System Dynamics Learning, MIT Media Laboratory. Available at https://dam-prod2.media.mit.edu/x/files/papers/Zuckerman_SystemBlocks_ISDC03.pdf

Horn, M.S., and Jacob, R.J.K., Designing Tangible Programming Languages for Classroom Use. *Proceedings of TEI'07 First International Conference on Tangible and Embedded Interaction*, (2007). Available at http://hci.cs.tufts.edu/tern/horn-jacob-tei07.pdf

Wyeth P, Purchase H (2002) Tangible Programming elements for young children. In: Extended abstracts from the proceedings of the CHI 2002 conference on human factors in computing systems, Minneapolis, Minnesota, April 2002. ACM Press, New York, pp 774–775. DOI 10.1145/ 506443.506591