MLTK01
A Prototyping Toolkit for Tangible Learning Things

Lupetti, M.L.; Romagnoli, Lorenzo

DOI
10.1007/978-981-19-4472-7_57

Publication date
2022

Document Version
Final published version

Published in
With Design: Reinventing Design Modes

Citation (APA)
Lupetti, M. L., & Romagnoli, L. (2022). MLTK01: A Prototyping Toolkit for Tangible Learning Things. In G. Bruyns, & H. Wei (Eds.), *With Design: Reinventing Design Modes : Proceedings of the 9th Congress of the International Association of Societies of Design Research* (pp. 883-889). (IASDR: Congress of the International Association of Societies of Design Research). Springer. https://doi.org/10.1007/978-981-19-4472-7_57

MLTK01: A Prototyping Toolkit for Tangible Learning Things

Maria Luce Lupetti[1(✉)] and Lorenzo Romagnoli[2]

[1] Faculty of Industrial Design Engineering, Delft University of Technology, Delft, The Netherlands
m.l.lupetti@tudelft.nl
[2] Automato Farm, Shanghai, China

Abstract. This work illustrates and reflects on the design process of MLTK01, an open-source toolkit for fast prototyping tangible learning things, built on top of Arduino and ml5js. The toolkit was developed as a response to the current lack of fast and easy to use tools for tangible experiments with machine learning. Learning from insights gained through previous projects, we defined a set of basic building blocks necessary to enable such experiments and engaged in an iterative process of sketching, prototyping and preliminary testing of the toolkit. MLTK01 includes a custom PCB, a software library and accessories. Together with a descriptive account of the design process we also discuss possible applications of the toolkit and its implications for a design process of tangible learning things.

Keywords: Machine learning · Toolkit · Tangible learning things · PCB · Fast prototyping

1 Introduction

With their growing availability and accessibility, machine learning (ML) algorithms are increasingly approached as a *design material* (Luciani et al. 2018; Dove et al. 2017; Yang et al. 2018(A); Yang et al. 2018(B)). These, not only are embedded into mundane products and services, e.g., spam filters in our email or personalized advertisement (Yang et al. 2018(B)), but starts to become widely available to non-expert users, both in research and practice environments (Sarkar 2016; Mellis et al. 2017). Both industry and academia, in fact, have been working towards lowering the barrier to use ML algorithms. From more technical approaches focused on improving ML interfaces and their user experience (e.g., Mellis et al. 2017) to hybrid approaches where ML tools are developed by multidisciplinary teams including designers (see the Magenta project by Google Brain (Kayacik et al. 2019)). As a result, there is today a large variety of freely available and easy to use ML tools, such as Trainable Machines, ml5js, Wekinator, Runaway ML and more. These offer novel opportunities for exploration and prototyping to creative practitioners (Fiebrink 2019). These tools are also increasingly being used to prototype *tangible learning things* both in design research (see the *The Morse Thing* project (Oogjes et al. 2020) or the project *Shybo* (Lupetti et al. 2017)) and professional contexts (see examples

G. Bruyns and H. Wei (Eds.): IASDR 2021, *[] With Design: Reinventing Design Modes*, pp. 883–889, 2022.
https://doi.org/10.1007/978-981-19-4472-7_57

from art and design practice like the *Objectifier* (Schwab 2017), *NORAA* (Visnijc 2018 (A)), *Soli* (Dufrense 2016), *Spot* (Aouf 2019) and more). Yet, tools specifically designed to support these tangible explorations are still lacking.

The prototyping process of ML tangibles is still characterized by multiple actions that require an intermediate level of expertise both in hardware and software development. The process usually starts with the selection of a ML model and of hardware components, which also need to be assembled. In order to prototype hardware that relies on software running on a computer (common practice when fast-prototyping things that embed ML abilities), thus, it is necessary to define a communication layer (wi-fi, bluethooth, serial) and a protocol (custom protocol, attention commands, open sound control), which need to be implemented on both the computer and the hardware. Ones the software and the hardware are connected, the ML model can be customized and used. Thereafter, this type of work still requires designers to create custom tools for bridging existing ML libraries and tools with hardware components, which may be a time-consuming activity and, for users with low technical skills, a discouraging challenge. To address this issue, we designed and developed MLTK01: a toolkit aimed at allowing creative practitioners to fast and easily prototype tangible learning things.

2 MLTK01 Toolkit

MLTK01 is a toolkit that allows to avoid or reduce hardware prototyping time and focus on the design of interactions with tangible learning things. The toolkit was developed by combining hardware and software from two existing open-source tools: Arduino and ml5js.

2.1 Board

Due to our intent of allowing users to easily plug-in the board, train and play ML sketches, we developed a configuration of hardware components (PCB) which, accompanied by illustrations printed on the custom printed circuit board (PCB), facilitate the flow of usage. The board comes pre-flashed with a custom firmware exposing the board bluethooth property and is intended to function in combination with a webpage running a sketch which includes the MLTK01 library. The essential hardware components consist of: connectors, a switch, a button, a rotary encoder, a LED ring and an Arduino nano BLE Sense. The latter was chosen especially to provide *manipulability*, as it comes with several useful components embedded: 9 axis inertial sensor, that can be used for developing wearable devices; humidity and temperature sensor, that allow to get highly accurate measurements of the environmental conditions; barometric sensor, which easily allow to make a simple weather station; a microphone, to capture and analyze sound in real time; and a color, brightness and proximity sensor that allows to estimate room's luminosity, as well as if someone is moving close to the board (Figs. 1 and 2).

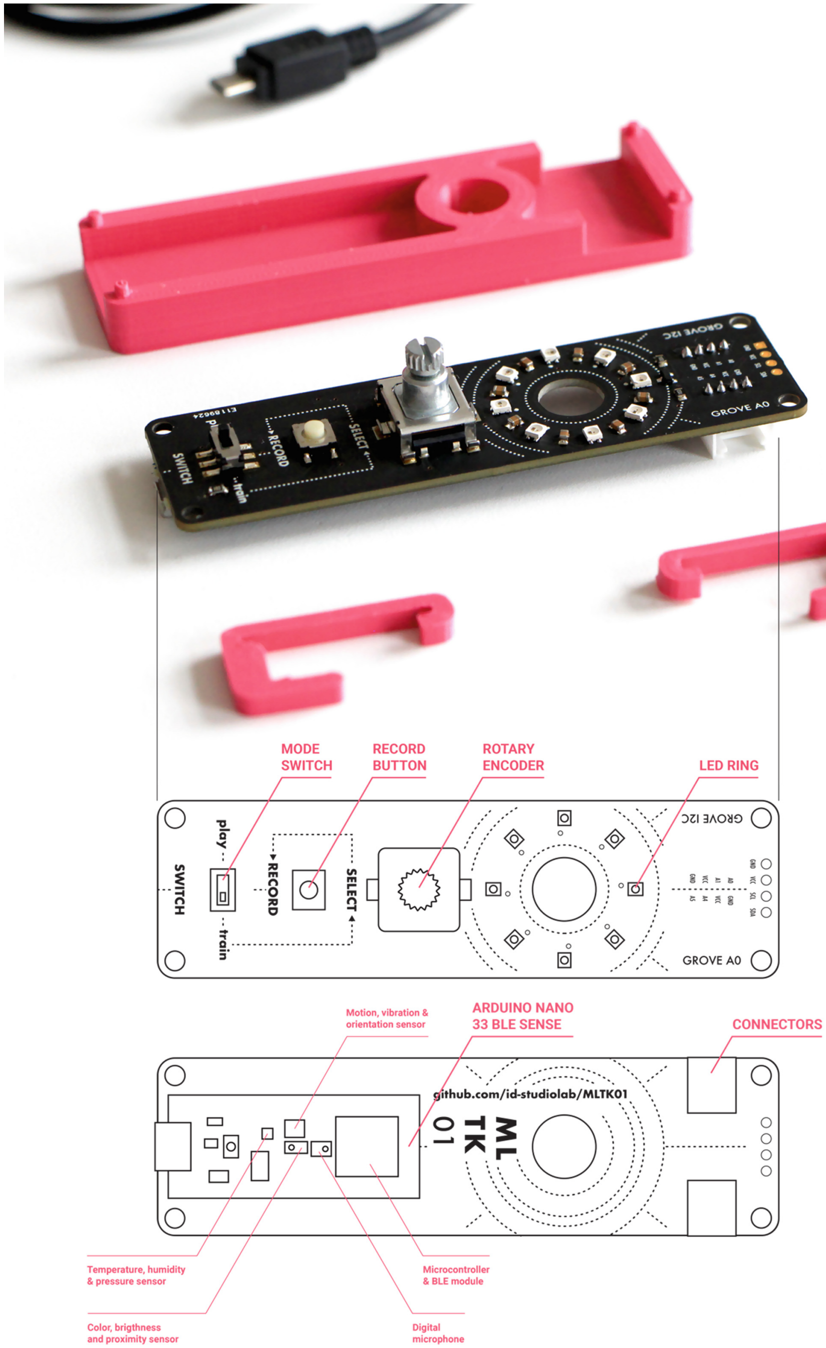

Fig. 1. MLTK01 board final prototype and components.

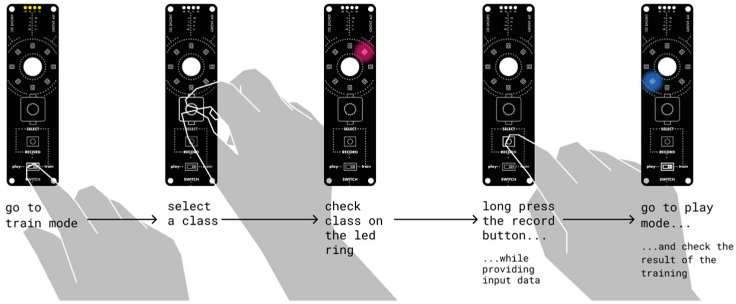

Fig. 2. Training sequence of the MLTK01 board.

2.2 Software Library

The MLTK01 library can be used directly into the p5.js[1] web editor. Before starting to explore the ML sketches, users can use a string of code available in the *getting started* section of the MLTK01 website to import the p5.js, the ml5js[2] and the MLTK libraries. This links the main sketch file and fetches a stylesheet designed to render some UI utilities coming with the MLTK library. The library includes four key components, that are: *MLTK object*, which access the features provided by the MLTK library; *train and play functions*, that switch modality from training to playing; *label variable*, which labels the collected data according to a selected class; and *features variable*, where can be specified the type of input data that are being collected.

3 Using MLTK01

Training the MLTK01 board consists of five steps: (1) *moving into train mode* by changing position of the switch; (2) *selecting a class* by turning the rotary encoder; (3) *checking the selected class* on the LED right; (4) *recording training data* by pressing the button; (5) *checking the results of the training* by switching back to play mode. If, when checking, the result is not as desired, more training may be needed. In that case, the user can simply switch back to train mode and record more input data.

3.1 Example Applications

The toolkit can be used to prototype a variety of different applications. Here we illustrate two alternatives, one making use of the tangible component as input, the other as output. The first example, *Simple learning robot*, uses an *Image Classifier* to train a little robot to recognize different people and react accordingly. The reactions are: calm, expressed through a light blue fading light and no movement; and alerted, expressed through a fast-fading red light and shaking movement. The second example, *Game controller* (Fig. 4), uses a *KNN classifier* to train the MLTK01 board to serve as game controller. Using the embedded 9 axis inertial sensor, in fact, the board can be trained to recognize different gestures which, then, can be used as input to control a game character actions (Fig. 3).

[1] https://p5js.org/.
[2] https://ml5js.org/.

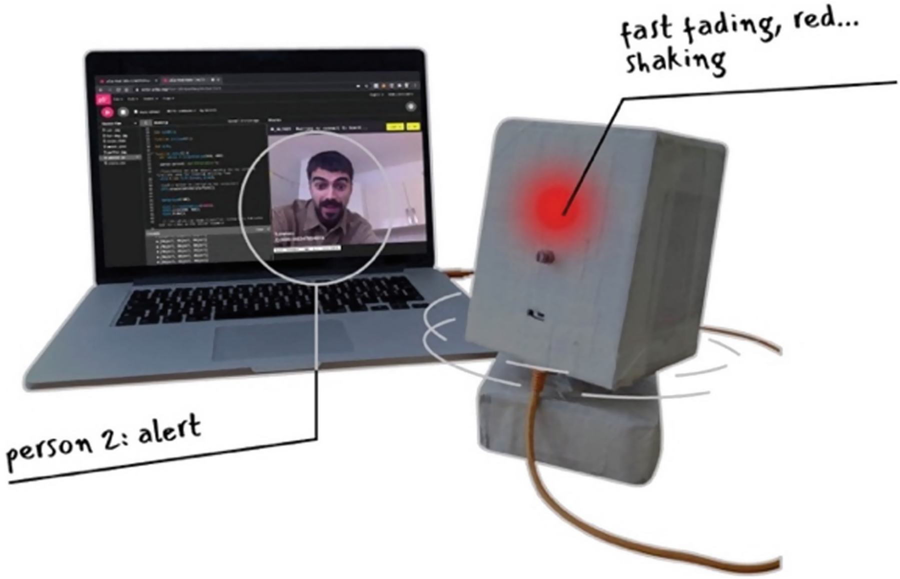

Fig. 3. Simple learning robot: reacts as scared or calm according to the person in front, trained using image classification.

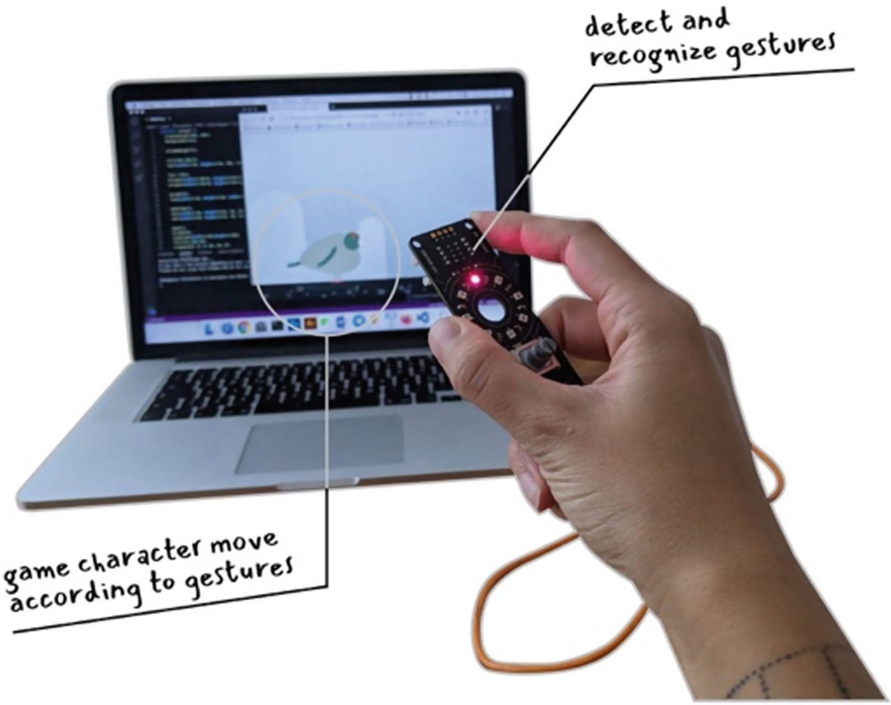

Fig. 4. Game controller: the board was trained to associate gestures to game commands by using a KNN classifier.

4 Conclusions

In order to understand interactions with ML interfaces, we looked at how ML processes are represented and enabled through existing tools. We, then, identified necessary hardware components that a board for tangible ML experiments should have and prototyped it. This was completed with a software library that combines p5.js, the ml5js libraries and a dedicated one. This way, we created a tool that can be easily set up and used for fast ML experiments without requiring hardware prototyping time and technical expertise. Initial hardware selection and assembly are not needed anymore: the user can customize the hardware by adding components but it is an optional expansion of the toolkit. Furthermore, MLTK01 allows to eliminate the need for developing a communication protocol and writing a custom firmware for the hardware used. Removing this process generates benefits for the user as it saves time, but foremost is a crucial benefit not only because saves time, but foremost because it lowers the barrier for designers with low technical expertise to engage with ML explorations. Thus, the examples shown on the previous page illustrating two options of how to operate the toolkit, demonstrate possible processes in which the toolkit can enable the development of tangible learning things. Our aim, however, is to challenge designers to explore, play and define meaningful ways in which materiality can be brought back into the explorations of ML, to challenge assumptions about what input data is and can be. In order to engage with this challenge, we will test, explore and use the toolkit as part of workshop activities including designers and artists to critically investigate the implications of ML as an enabling system to allow for speculations in regards to emerging relationships between human and artificial intelligence in the near future.

Acknowledgments. We would like to thank Elisa Giaccardi for her support and advice. We would also thank the Delft Institute of Design for Values for supporting the project through the DDFV seed project fund.

References

Aouf, R.S.: Spot toy by New Deal Design encourages kids to be “in-the-moment”’. Dezeen (2019). https://bit.ly/2ZH6bhY. Accessed 16 Sep 2020

Dove, G., Halskov, K., Forlizzi, J., Zimmerman, J.: UX design innovation: Challenges for working with machine learning as a design material. In: Proceedings of the 2017 Conference on Human Factors in Computing Systems, pp. 278–288 (2017)

Dufrense, S.: World’s tiniest violin uses radar and machine learning. Hackaday (2016). https://bit.ly/3kgausy. Accessed 16 Sep 2020

Fiebrink, R.: Machine learning education for artists, musicians, and other creative practitioners. ACM Trans. Comput. Educ. (TOCE) **19**(4), 1–32 (2019). https://doi.org/10.1145/3294008

Fiebrink, R., Cook, P.R.: The Wekinator: a system for real-time, interactive machine learning in music. In: Proceedings of the Eleventh International Society for Music Information Retrieval Conference, Utrecht, NL (2010)

Kayacik, C., Chen, S., Noerly, S., Holbrook, J., Roberts, A., Eck, D.: Identifying the intersections: User experience + research scientist collaboration in a generative machine learning interface. In: Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, UK, pp. 1–8 (2019). https://doi.org/10.1145/3290607.3299059

Luciani, D.T., Lindvall, M., Löwgren, J.: Machine learning as a design material: a curated collection of exemplars for visual interaction. In: DS 91: Proceedings of NordDesign 2018, Linköping, Sweden (2018)
Lupetti, M.L., Yao, Y., Mi, H., Germak, C.: Design for children's playful learning with robots. Future Internet **9**(3), 52 (2017). https://doi.org/10.3390/fi9030052
Mellis, D.A., Zhang, B., Leung, A., Hartmann, B.: Machine learning for makers: interactive sensor data classification based on augmented code examples. In: Proceedings of the 2017 Conference on Designing Interactive Systems, Edinburgh, UK, pp. 1213–1225 (2017). https://doi.org/10.1145/3064663.3064735
Oogjes, D., Wakkary, R., Lin, H., Alemi, O.: Fragile! handle with care: the morse things. In: Proceedings of the 2020 ACM Designing Interactive Systems Conference, Eindhoven, NL, pp. 2149–2162 (2020). https://doi.org/10.1145/3357236.3395584
Sarkar, A.: Constructivist design for interactive machine learning. In: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, San Jose, USA, pp. 1467–1475 (2016). https://doi.org/10.1145/2851581.2892547
Yang, Q.(A)., Suh, J., Chen, N.C., Ramos, G.: Grounding interactive machine learning tool design in how non-experts actually build models. In: Proceedings of the 2018 Designing Interactive Systems Conference, Hong Kong, CN, pp. 573–584 (2018). https://doi.org/10.1145/3196709.3196729
Yang, Q.(B)., Banovic, N., Zimmerman, J.: Mapping machine learning advances from hci research to reveal starting places for design innovation. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Montreal, CA, pp. 1–11 (2018). https://doi.org/10.1145/3173574.3173704
Schwab, K.: If You've Trained A Dog, You Can Train This AI. Fast Company (2017). https://bit.ly/35F9kCR. Accessed 16 Sep 2020
Visnijc, F.: (A) NORAA (Machinic Doodles) – A human/machine collaborative drawing (2018). Creative Applications. https://bit.ly/2FGXhKx. Accessed 16 Sep 2020