Configurable Tangible Programming Based on Event Programming for Interactive Project

Budsapanee Pongsiriyaporn [1], and Sakol Teeravarunyou [2]
[1][2] School of Architecture and Design
King Mongkut's University of Technology Thonburi, Thailand
Corresponding author: budsapanee@gmail.com [1] , sakol.tee@mail.kmutt.ac.th[2]

ABSTRACT

Programming is an essential skill for the future workforce in controlling smart homes, intelligent products, interactive products, robots and automation. In fact, not so many users have learned programming before. From the previous studies, tangible programming is found to increase effectiveness on novice users. They can program by connecting physical building blocks together to control interactive products without having to connect to any personal computers. The problem of existing tangible programming is that it cannot be used for the complexed tasks. The aim of this research is to develop a new tangible programming based on event which is more appropriate for sophisticated tasks. The new design hardware named C-TAN which includes two main parts of device hub and command block. Intermediated users tested with the existing (LittleBits) and event tangible programming (C-TAN). The results show that the learning curve of Littlebits is better than C-TAN in easy and normal tasks. Nevertheless, the C-TAN satisfies users more on the difficult tasks by using less modules comparing to existing ones. It also benefits in terms of flexibility of managing modules and enhancing customization.

KEYWORDS: Tangible programming, Event programming, Learning curve

1. INTRODUCTION

Since the first computer was created, everything has been developed rapidly. Labors are being replaced by technology in this day. Machines can do more precision work with less error. Foxconn, the manufacturer of electronic parts, layoffs workers around 60,000 positions and use automation system instead but Foxconn still need workers who maintain the automation system including the researchers (Wakerfield, 2016). The human needs to develop themselves for the new opportunity in the age of AI. 80 years ago, a computer was a huge machine for a specific purpose. It was created to serve the military. It had been reserved for experts and scientists only. At present, a computer can be used for multiple purposes and it is so small to fit your hand, for example, smartphone, tablet, and laptop. Until Internet was created and used widely, the era of digital was truly begun and it will be the long way to go.

Many intermediated users use tools in the market for several applications. For example, the lighting designers use DMX to control the light of backstage. Users need to use with the lighting software, for example, Elation Emulation DMX Software/Hardware Lighting Controller, Blizzard Lighting Eclipse DMX Controlling Software, Sunlite Suite 2, MADRIX Software and others. While exhibition designer uses a microcontroller such as Arduino to control the electronic equipments. One of the popular tools to do projects is Arduino. Arduino is a microcontroller has been used to program and control input devices, output devices and sensors. The Arduino software (IDE) was developed from Processing. There are a lot of compatible languages, for example, ArduBlockly, Snap4Arduino, C# and Python to program on Arduino board (Jones, 2015). Although several tools are provided, they cannot serve the non-programmer fields. For example, exhibition designers have less experience of programming but they want the results that can programable.. The goal of intermediated users is to achieve the solution with less programming.

1.1 Statement of problems

Wiedenbeck suggested that the knowledge organization is important to develop programming skill because programming need to think step by step with the systematic process (Wiedenbeck, 2005). According to Wiedenbeck's study, novice programmers should not sudden leap in difficulty because they develop their knowledge organization when they learn to code and practice slowly step by step. Chang suggested that the computer anxiety relate to complexity of programming task in teaching and learning. The increasing programming experience help reduce the anxiety when novice programmers are coding (Chang, 2005).

Similar to the intermediated programmer, they have some background from fundamental programming but they cannot do the complexed tasks. The requirement of depth knowledge as experts is still a problem of learning. The objectives of this research is to design a tool that can help intermediated users achieve their goals with small learning step. The tangible programming is another solution that intermediated users are able to use without dealing with the complexity of codes. Tangible programming is a tool to help and encourage people to learn programming easily. Most of tangible projects were test with children and elderly but not the working age. These people have problems about programming, especially designers who need to program the hardware for their interaction design. This point drives us to develop the tool that can program without the requirement of advanced knowledge of programming. Designers just program by interacting with objects instead of software.

2. LITERATURE REVIEW

At present, technology infiltrates to everyone. It tends to be 'Do It Yourself' era in the future. Programming is a psychomotor skill that requires time to practice like other skills such a painting, cooking and bicycle riding. There are current research that addresses the tangible programming.

2.1 Tangible Programming

The concept of the hand-mind connection is about implication of motor skill and brain. Jean Piaget suggested that intelligence grows from the interaction of the mind with the world. There are so many works reveal that tangible programming help users understand quickly and easily, for example, TurTan was developed by Gallardo et al. (2008). TurTan is Tabletop, projector and camera under table to capture activities on surface, identify and detect tangible objects and finger in real-time. The result revealed that TurTan help users understand quickly, explore easily and encourage more about social interaction including collaboration. For the collaboration learning, Suzuki et al.,(1993) revealed that AlgoBlock encourages team collaboration. They are blocks with commands such as go forward, slide to left/right, turn and others. In addition, control blocks such as if-then-else and repeat-until are available for logical control. It is not only for the children but also the elderly too. Tangibot (Sanjuan et al., 2016) helps elderly in many ways such as human-to-human socialization, bring more natural and intuitive interactions. Evaluating children performance with graphical and tangible programming tools used Proteas (PROgramming Tangible Activity System) was developed by Sapounidis et al., (2015). It is cube-shape blocks that represent basic and advance programming structure. It is used to command the robot.

If we assumed that designers do not have programming knowledge, Wyeth and Wyeth's (2001) study would be useful for our research. It is electronics blocks that contain circuit, for example, sensor blocks, action blocks and logic blocks inside. Children can stack and arranged the blocks to create different functions such as light and sound. Horn and Jacob (2007, 2008) suggest Designing Tangible Programming Languages for Classroom Use or Quetzal to help children learn about complicated syntax easier, encourage collaborative and keep positive learning environment by arranging parts on the portable scanning with image processing technique to compile the code. TanProStory (Qi et al., 2015) was developed to teach Object Oriented Programming concept through storytelling. It consists of three parts: Programming blocks (used to create characters and construct programming sequence), Animation game (provides a storytelling environment for children and acts as processing center and output) and Sensor input module (small box allowing children to trigger different kinds of sensors to make it more interesting). The result shows that most children can complete the tasks with naturally and full of motivation. Some of children do it wrong, after got the feedback they know what is the mistakes and able to correct by themselves. Moreover, TanProStory help children learn about event handling concepts and sensors also some traffic rules. That is very similar to our objective to help our users handle their projects and show the results of programming. McNerney suggested tangible programming bricks for making programming accessible to everyone (1999) and exploration in physical language design (2004). Tangible programming bricks are electronic toys which help children develop advanced modes of thinking through free-form play, construction toys and hand-on learning to explore computation and scientific thinking. His studies revealed that tangible program is easier to understand, remember, explain to other, and perform in social setting than traditional programming.

Based on the reviews, it is the truth that tangible programming is very easily to understand quickly and it encourages collaboration but there are gaps too. The first gap is the result from almost studies used tangible programming to create input and almost outputs were appeared on the screen. It is not proper to create rapid mock up in real life because it was created for teaching not to build the actual things. Fernaeus and Tholandetry (2006) try to improve and reduce the distance between physical and virtual representations. They find the design qualities in a Tangible programming and their studies revealed a good coupling between physical and virtual actions in the certain points of interaction. In addition, Blackwell (2003) suggested the cognitive dimension of tangible programming languages. He tried to explore more about generic possibilities of tangible interfaces for programmable functionality and the target is to develop query language that can be used in school based on physical elements with RFID tags. The Digital Dream Lab (Oh et al, 2013) is tabletop puzzle block for exploring programmatic concepts. This set consists of puzzle blocks (e.g. character, animation, color, and size) tagged with ReacTIVision, interactive tabletop with camera situates under the table to detect identity, position, orientation, and projector for projected them on vertical surface. Tada and Tanaka (2015) suggest the tangible programming environment using paper cards as command objects. It is paper cards called Sheet (as command objects), webcam, and software, to create the program, users need to arrange the sheet in an order then the webcam captured, recognized by software and displayed the graphic on screen. These sheets are special because users can draw more on it to edit the program. This is would be the second gap because it is suitable for children to learn but it is not suitable for people who need to program in short-time. These are too fundamental for adults. We think that our target users already know basic knowledge but still struggle with complexed programming language. The evidence was appeared in Goran Zaharija's work (Zaharija et al, 2014). They try to encourage deep logical thinking and more immersive experience. The third gap is that it cannot create sophisticated programming. Some study used tangible programming as tools for controlling such as Dr. Wagon (Chawla et al, 2013). Dr. Wagon is a robot and series of programming blocks with basic functions (move), conditions (if), and loop (repeat) to control the behavior of Dr.Wagon. According to previous studies, tangible programming is good to develop logical thinking in children, but it is not practical to create the actual projects.

2.2 Existing solution

LittleBits is a finished product in the market that has been created for children and beginner (Bdeir, 2016). A user can and play intermediately. It is not required programming skill and hardware knowledge to use it. The magnetic connectors of each block can control the direction when a user plug them together and prevent the short circuit problem. It consists of small circuit boards with specific functions built to snap together with magnets without soldering or programming (see Fig. 1). Each bit has its own specific functions such as buttons, LED light, and sound. One of the problems in LittleBits is about the structure of programming. Since each piece of module connects as a series circuit, it has limitations in terms of programming. The logic of programming will follow like the command line of building blocks (linearity) that cannot be customized for the advanced programming. For example, users cannot blink the light automatically.

C-Tan is an event-handler concept that was designed to help intermediated users. It can be used to control the sensors and actuators such as LEDs, motor and sound. The good thing about this device is that the behavior of devices was designed to accommodate the different types of projects. It consists of two main bodies: device hub and command block (see Fig. 2). The device hub is for input and output modules while the command block is for behavior such as the smoothness, brightness, and time duration of aculators. As a result, stage lighting designers can control the behavior of lighting by using the command block.

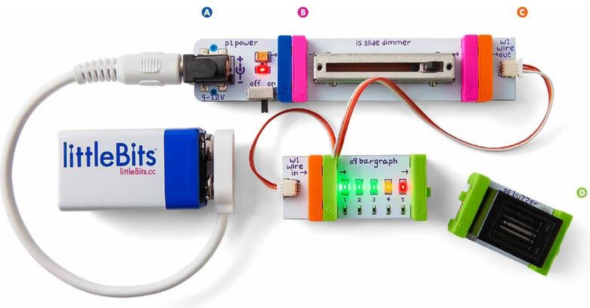

Figure 1. LittleBits

2.3 The Design of C-TAN

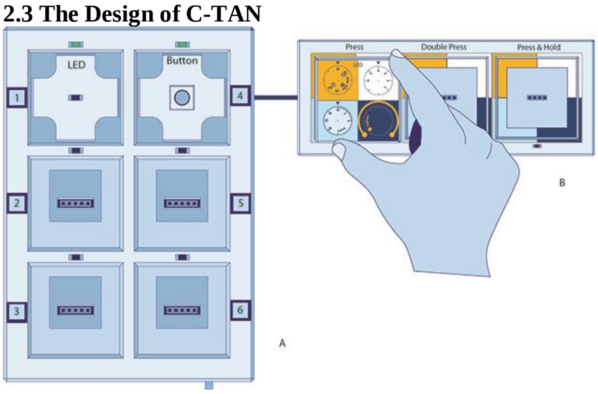

Fig 2. C-TAN with a) Device hub and b) Command block

2.3 Context of users

2.3.1 Existing users

Our existing users or target users is novice programmer or designer who would like to create simple hardware project. After we interviewed some designers for preliminary study, they revealed that they know what is processing to get the result that they want but programming knowledge is probably huge barrier for them. Moreover, they need to know about hardware programming to make their projects work.

2.3.2 Potential users

DIY culture has been started since 1910. At that time, people favor to fix or invent something when they free. The spirit of the maker was transferred to today. Nowadays, everyone can be a maker. They can create products by using 3D printer and laser cutting machine. With a microcontroller, they can build several projects. They do not need to be engineer or technician since there are so many tutorial on the social media such as Youtube and Internet.

2.4 Research objective

There are two objectives of this study :

1. To design the tangible programming that could be possible to control the output devices.
2. To compare the learning curve between event and sequential programming in term of goal achievement.

2.6 Scope and limitation

Since there is not so much tangible programming product in the market, the LittleBit will be selected to compare with C-TAN. Although the LittleBits may not be used for the experts, it is potentially used for intermediate users. LittleBits can extend its capacity to build a simple robot arm and painting arm.

3. METHOD

3.1 Experiment

Fig 3 (A) C-TAN Experimental and (B) C-TAN Experimenta*l*

Five freshmen who are an intermediated users of computer programming participating in the experiment. All of them had a basic skill of programming before. They were assigned to do 9 tasks (A to I). Task A-C is an easy task. Task D-F is normal and G-I is difficult task. It was setup to measure the learning curve. The average time of experiment is 30 mins to 1 hour. The scale of 0 to 4 with 0.5 level in between. Table 1 shows the task from A to I. Task A,B,C is easy task. Task D, E, F is normal task. Task G, H, I is the advanced task (see Table 1). Table 2 is a level of achievement that can be used to score the participants.

Table 1. Task list for experiment

Task	Description	Level
A	Turning on LED	Ease
B	Turning on buzzer/speaker	Ease
C	Blinking LED	Ease
D	Dimming LED	Normal
E	Changing the tone level of piezo/speaker	Normal
F	Making LED and piezo/speaker work simultaneously.	Normal
G	Turning LED on when the environment is dark. Then playing sound when switch was pressed.	Difficult
H	Turning on LED when something passes.	Difficult
I	Turning on LED when the switch is pressed once and playing the sound of buzzer/speaker when the switch is pressed twice.	Difficult

Table 2. Levels of achievement

Score	Programming skill
4.0	Accomplish the task without assistant. Understand how to use the device correctly and less error.
3.5	Be able to finish the task without assistant. Still have error (1-5 times per task).
3.0	Be able to finish the task without assistant but they still have some error around 6-10 times per task.
2.5	Be able to finish the task without assistant but they still have some error more than 10 times per task.
2.0	Can finish the task but need assistant 1-3 times. Cannot understand how to use device perfectly and still have a lot of error.
1.5	Can finish the task but need assistant 4-6 times (asking or manual). Cannot understand how to use device perfectly and still have a lot of error.
1.0	Can finish the task but need assistant 7-12 times (asking or manual). Cannot understand how to use device perfectly and still have a lot of error.
0.5	Can finish the task but need assistant by using manual or asking. Still cannot understand how to use device and have a lot of error.
0	Cannot finish the task. Need assistant.

4. RESULTS

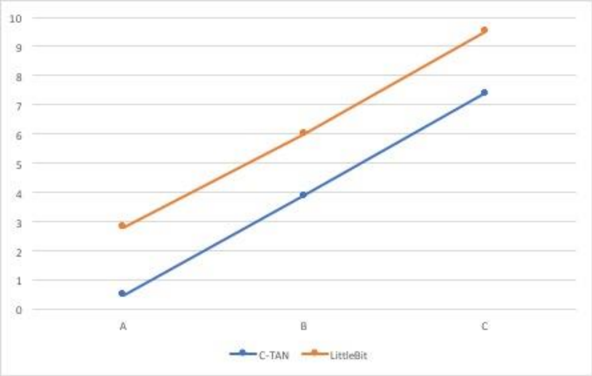

Fig 5. Average scores of task A to C

The learning curve is done by measuring the cumulative score of each trial. Figure 5 shows an average of task A to C. The result revealed that LittleBits (Sequential building block) has higher score than C-TAN (event based programming) around 2 points. The reason is that LittleBits is easy to use because of interface design. It has less steps when operating the hardware. LittleBits is difficulty to do in some task such as blinking task (Task C). C-TAN takes an advantage of automatic control. They can control the behavior of blinking based on time duration.

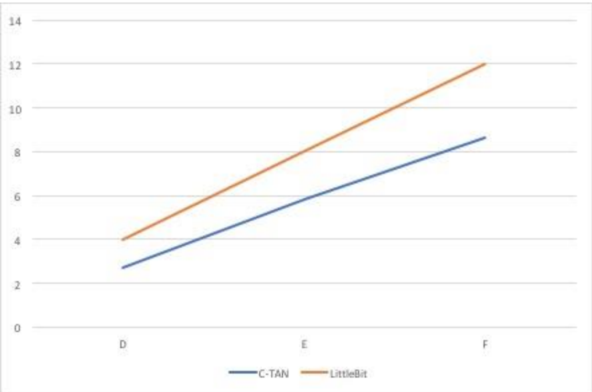

Fig 6. Average scores of task D to F

For the normal tasks (Task D,E,F), the learning curve of event based programming was dropped but the sequential building block is very high (see Fig. 7). The reason is that LittleBits can be manipulated easily with the control modules. Similar to Task C, LittleBits cannot do the dimming light (Task D) automatically. Another problem of LittleBits is the reuse of the same device, for example, one button cannot do many functions. C-TAN has an advantage over this point.

For the advanced task such as Task G, H, I, the learning curve of event based programming is higher than the sequential building block (see Fig. 7). C-TAN is flexible when operating many output devices than LittleBits.

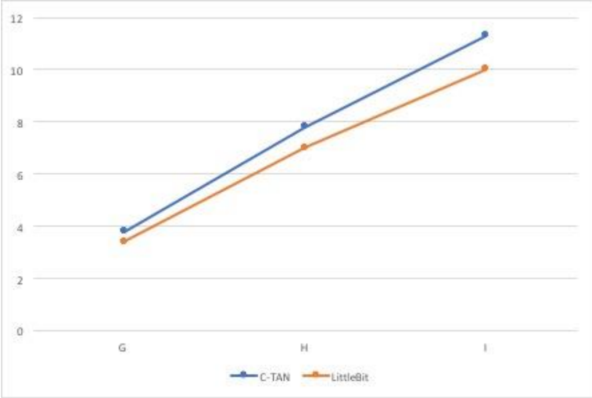

Fig 7. Average scores of task G to I

Both learning curve of LittleBits and C-TAN looks more linear than exponential curve (a long time to learn). They cannot reach the stable point. The suggestion is to find the tasks that have the same levels of both tools.

5. DISCUSSION

After testing, the LittleBits fits to the easy and normal task, while the C-TAN fits to the difficult task. The learning curve of LittleBits cannot extend to the difficult level, while C-TAN can accommodate all tasks from simple to difficult. The down side of LittleBits is a magnetic connection that is not stable much so the current cannot run through the whole circuits if all modules are not aligned together. While the C-TAN has a shortcoming on connectors between modules. Subjects must precisely plug-in the devices on the device hub. Another benefit of C-TAN over LittleBits is the automatic control of the behavior, for example, C-TAN can control the number of time for dimming the light. As a result, this feature can shorten the learning time. C-TAN also is good in term of configuration. It does not need to be rearranged the modules as same as LittleBits.

Event programming may take time to learn at the beginning since subjects are familiar with the learning method of series programming before. In the long run, subjects prefer the event programming more than sequence programming since they can customize the usage. The learning curve is high when using the event programming to handle the difficult task.

The further development of C-TAN is to redesign the connection and reduce an unnecessary functions. The affordance of C-TAN may need to be improved to fit the different types of command devices. For example, the lighting device may not the same as the sound device.

REFERENCES

Bdeir, A. *U.S. Patent No.9831599B2*. Washington, DC: U.S. Patent and Trademark Office. 2016

Blackwell, A., Cognitive dimensions of tangible programming languages, *In Proceedings of the First Joint Conference of EASE and PPIG*, pp. 391-405, 2003.

Chawla, K., Chiou, M., Sandes, A., & Blikstein, P., Dr. Wagon: a 'stretchable' toolkit for tangible computer programming, *In Proceedings of the 12th International Conference on Interaction Design and Children*, ACM, pp. 561-564, 2013.

Fernaeus, Y., & Tholander, J., Finding design qualities in a tangible programming space. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, pp. 447-456, 2006.

Gallardo, D., Julià, C. F., & Jorda, S., TurTan: A tangible programming language for creative exploration. *In Tabletop.*, pp. 89-92, 2008.

Goran Zaharija, Sasa Mladenovic, Ivica Boljat, Introducing basic Programming Concepts to Elementary School Children. *Procedia - Social and Behavioral Sciences.*, Vol. 106, pp. 1576-1584, 2014.

Horn, M. S., & Jacob, R. J., Designing tangible programming languages for classroom use. *In Proceedings of the 1st international conference on Tangible and embedded interaction., ACM*, pp. 159-162, 2007.

Horn, M. S., Solovey, E. T., & Jacob, R. J., Tangible programming and informal science learning: making TUIs work for museums, ACM , pp. 194-201, 2008.

McNerney, T. S., Tangible programming bricks: An approach to making programming accessible to everyone Doctoral dissertation, *Massachusetts Institute of Technology*, 1999.

McNerney, T. S., From turtles to Tangible Programming Bricks: explorations in physical language design, *Personal and Ubiquitous Computing*, 8(5), pp. 326-337, 2004.

Nacher, V., Garcia-Sanjuan, F., & Jaen, J., Evaluating the Usability of a Tangible-Mediated Robot for Kindergarten Children Instruction, IEEE, pp. 130-132, 2016.

Oh, H., Deshmane, A., Li, F., Han, J. Y., Stewart, M., Tsai, M., ... & Oakley, I., The digital dream lab: tabletop puzzle blocks for exploring programmatic concepts, ACM, pp. 51-56, 2013.

Qi, Y., Wang, D., Zhang, L., & Shi, Y., TanProStory: A Tangible Programming System for Children's Storytelling, *In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, ACM, pp. 1001-1006, 2015

Sapounidis, T., Demetriadis, S., & Stamelos, I., Evaluating children performance with graphical and tangible robot programming tools., *Personal and Ubiquitous Computing*, *19(1)*, pp 225-237, 2015.

Sheeson E. Chang, Computer anxiety and perception of task complexity in learning programming-related skills, *Computer in Human Behavior*, Vol. 21, pp. 713-728, 2005.

Susan Wiedenbeck. Factors affecting the success of non-majors in learning to program. *ICER '05 Proceedings of first international workshop on Computing education research*, ACM, pp. 13-24, 2005.

Suzuki, H., & Kato, H., Interaction-level support for collaborative learning: AlgoBlock—an open programming language. *In The first international conference on Computer support for collaborative learning, L. Erlbaum Associates Inc.*, pp. 349-355, 1995.

Tada, K., & Tanaka, J., Tangible Programming Environment Using Paper Cards as Command Objects, *Procedia Manufacturing*, 3, pp.5482-5489, 2015.

Wakerfield, J. Foxconn replaces '60,000 factory workers with robots', BBC News, Access on March 26, http://www.bbc.com/news/technology-36376966, 2016.

Wang, D., Qi, Y., & Zhang, L.. A tangible programming system conveying event handling concept. *In Proceedings of the 14th International Conference on Interaction Design and Children*, ACM, pp. 319-322, 2015.

Wyeth, P., & Wyeth, G. F., Electronic blocks: Tangible programming elements for preschoolers. *In IFIP TC. 13 International Conference on Human-Computer Interaction*, IOC Press, Vol. 1, pp. 496-503, 2001.