A Tangible Programming Language for Educational Purpose

In recent years, learning computer programming and developing computational thinking is becoming a trend in children's education. Various education-oriented tangible programming products emerge in endlessly, and openness and scalability have always been the core ideas of product design. In this report, our group have designed an education toolkit for children, which uses physical blocks to encapsulate C language programs, while designing visualized icons to help users understand. Children can use the product to realize different functions by combining different physical blocks according to a certain logic, which meets the requirements of openness. In addition, the product has a simple and exquisite appearance, which is easy to carry and convenient for users to program anywhere. Tangible programming products can help children learn and develop programming thinking through games, which combines fun and education. This report is divided into five parts. The first part introduces the background and understanding of tangible programming, combined with providing the general overview of proposal. Secondly, it illustrates the design of tangible programming language from three cases to show the interaction with user and issues, which is music, light and car, followed by describing the implementation of the system. Next, it shows a clear evaluation by inviting 14 children and conducting a survey on the difficulty and satisfaction of using the product, which bring more inspiration about the product optimization. Finally, the conclusion section proposes the broader application and summary of the results, followed by describing individual contribution to the project.

CCS CONCEPTS • **Human-centered computing → Human computer interaction (HCI)**

Additional Keywords and Phrases: Tangible UIs, education, children, programming languages

1 INTRODUCTION

programming is an abstract activity that requires developers to spend a lot of energy and time in writing code, running, and debugging in an obscure programming language. However, the increasing importance of programming has given birth to the ideas of education and technology workers who want to make it more interesting and understandable. Thus, tangible programming has been created.

Learning the background of tangible programming bring a deeper understanding of it. Suzuki and Kato's Algo Blocks is an early example, where the interlocking aluminum blocks represent language commands similar to Logo. In addition, McNerney's Lego blocks with embedded microprocessors can also express programming languages. Besides, Scratch is commonly used by students, which is developed by MIT Media Lab, and the users can execute the algorithm by combining different blocks in sequence. [1]

It can be seen that tangible programming is a language similar to text or visual programming languages, but instead of using text or pictures on a computer screen, it uses physical objects to represent different programming elements, commands, and control flow structures. The encapsulation of different functions and physical form make people can better understand the principles of programming. Additionally, the main target group of tangible programming is children, and it not only teaches children the concept and logical way of programming, but also cultivates children's computational thinking. Besides, it makes excellent use of

children's playful nature to stimulate their interest in programming and bring different programming experiences.

Therefore, we designed a tangible programming product named Tangible Program Blocks, which is an educational toolkit composed of code blocks and actuator. In older to achieve different functions, users can combine the packaged code modules according to logic to control actuator. For code blocks, except the output blocks, the introduction of structure blocks like branch and loop make children can better understand the programming language and logic. Besides, actuator include lights, motor, servo and some other devices, which can give users feedback on the success of the program through the light emission and the movement of the motor and servo, so as to give children a better sense of experience and increase their interest in programming.

2 DESIGN

This project is an educational toolkit for primary and middle school students to learn the thought of programming without being trapped by codes. This session will introduce main concept of the tangible program blocks, interaction with user, and issue addressed.

2.1 Main Concept of the System

This project is mainly used for students to learn programming, in which physical blocks and actuators are involved. Physical blocks, or called code blocks, are consist of circuit boards with C codes encapsulated, filling materials like sponges, and printed icons. Actuators are peripheral devices such as LEDs, buzzers, motors, and so on.

Students may have abundant remarkable imaginations, which should not be limited by the educational mediums. Hence this product just did limited encapsulation of codes so that it provides more possibilities for students to create new meanings. The new language is described in "the Programming Language" session.

2.1.1 The Blocks

As previously stated, the inner content of each block is a circuit board, on which there is a core chip, a LED that signs the status of the board, and connectors that used for connecting to other boards or devices. According to built-in codes in the chip, a specific functionality is assigned to a board.

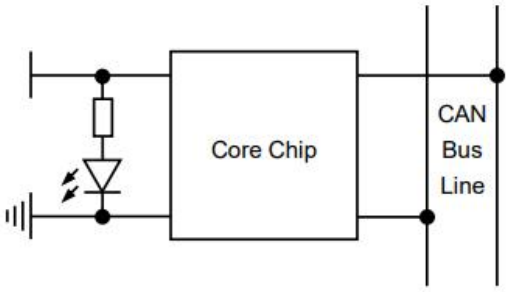

Figure 1: Design of Circuit on Board

To connect the boards to each other, on-board connectors are used. Design was made while selecting the components. Through the connectors, not only power can be supplied, but data can also be transferred. One way to organize the blocks while coding is to make them directly connect to one another with the connectors. If a program has a certain scale and some blocks are too far to connect directly, the on-board connectors can be linked by jumper wires.

When connected, the chips will take advantage of CAN protocol and establish a CAN bus to communicate with each other, in order to know what the whole program is. Then user can link the "begin" block, on which there is a USB connector, to another device and transfer the complied program. Such device could be a LEGO Mindstorms main block like RCX, NXT, or EV3. Or it could be other board or chip that runs C program, like an Arduino or another MCU.

2.1.2 The Programming Language

As indicated earlier, it is necessary to protect children's ideas from being limited. Thus, this project is designed under the idea that providing tools basic enough so that students can use them to create nearly whatever they imagine. Therefore, the programming language of this project is just a low level encapsulation of the C language. Consequently, this language inherits the advantages of C, and provides students with enough freedom and less limitation to program for diverse platforms. Meanwhile, this language hides the complex details of C, so it reduces the cost of learning programming.

While designing the language, three fundamental aspects were focused on: structure, input, and output. [2]

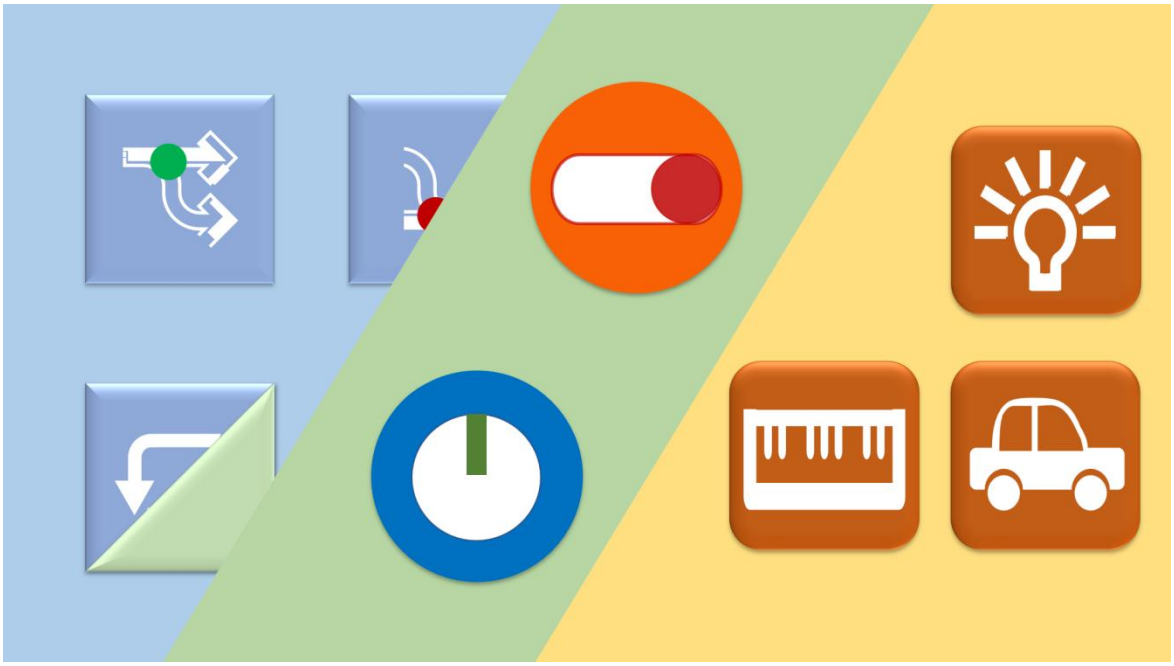

Figure 2: Examples of Structure, Input, and Output Blocks

Regarding structures, sequence is basic. In terms of selection, there are "fork" block and "merge" block. Students will learn how to decide which branch of program to run. As for repetition, there are "start loop" block and "end loop" block. The program covered by these two blocks will be repeatedly executed. It does not cost much to encapsulate the "goto" statement of C. However, it is not recommended. [3]

Besides, there are also "begin" block and "end" block that signs the beginning and the end of a program.

As regards input blocks, they are mainly used to take control of sensors and collect data. The collected data could be used in other parts of the program, like conditions for selections and repetitions.

With regard to outputs, the build-in code of blocks has called, realized, or encapsulated the common functions for actuators. Thus, students will not get troubled by GPIOs, analog quantities, PID, and so on. Hence, children can focus on their train of thought, and establish a preliminary understanding of actuators.

2.2 Interaction with User

While designing the interaction, efforts were made to gain deeper understanding of the materials and better interaction between users and the product. Universal design principles are followed to broaden the range of users. On the blocks figurative icons are used to make the project simple and intuitive to use. Every block consists of color, shape, and images with abundant meaning, which make the project regardless of the knowledge, experience, language or level of concentration of the user.

The major target audience, students, may have limited vocabulary, and still learning grammars of natural languages. Thus, codes and syntax of programs may be confused. So, blocks with images can help students understand the idea of programming.

Perceptible information is provided. On a circuit board there is a tricolor LED that signs the status of the board. The condition of the programming process could be told from the color and flicker of the LED:

- Green: the program is successfully built.
- Red: the connection is inappropriate.
- Yellow: there is a compilation error (the block lighting yellow can be considered as a break point).
- Slow flicker: a connected to a peripheral device
- Fast flicker: data is transmitting between the block and the device.

2.3 Issue Addressed

People are diverse. So the idea of multi-sensory is adopted to satisfy special needs. Meanwhile, disabled person can also be involved to learn programming in this language.

On the surface of blocks, diverse colors are used to distinguish different categories of functions of the blocks, including structures, sensors, behaviors, and so on. Colors with high saturation are used for learners with color weakness. Also, patterns like dots and lines are drawn on the surfaces related to their color for color-blinded users.

There are also physical grains and textures of those patterns, so that people can recognize the blocks when touching them without seeing them.

Additionally, considering younger children, the base of blocks is designed in size of 5cm * 5cm, in aim of easier catching. Fillets with radius of 1cm is designed, and the pins for connection are hidden and wrapped by soft sponge, in order to avoid potential injuries.

2.4 Stakeholders

Since the tangible program blocks is aimed for children, so the primary stakeholders are children.

The secondary stakeholders are customer, development company management, training course management. In regard to the tertiary stakeholders, they are competitors, parents. In addition, the facilitating stakeholders are team 2, IT staff. The design team has five members, and each of them has different roles. Hao Yukun is responsible for the design of blocks, Peng Yibo is the designer of languages. In addition, Nan Jing takes the role of designing icons, buttons and interface, Wang Jiawei makes contribution to the demo and the design of car. Last but not least, Guo Jie is responsible for the idea of our product.

3 IMPLEMENTATION

Since there is no real built-in code for the circuit board, we choose to use the Wizard of Oz technique.

In the process of implementation, we prepared the basic tools and bought some papers, a 1m*1m sponge and ten 5cm*5cm circuit boards with designed circuits.

Paper is cheap and convenient, which makes it easy to design.

Sponge can not only increase the thickness of the module, create a three-dimensional effect, but also can be used as external packaging for protection.

Moreover, the circuit boards, not only can be customized the size, but also can better imitate the prototype appearance.

Coloring pencils are used to draw three types of functional modules in different color and shapes, which is structure, input and output blocks: structure, input and output modules. Then cut the corresponding sponges, attach the module labels to the sponges using solid glue and then glue the whole thing to the board.

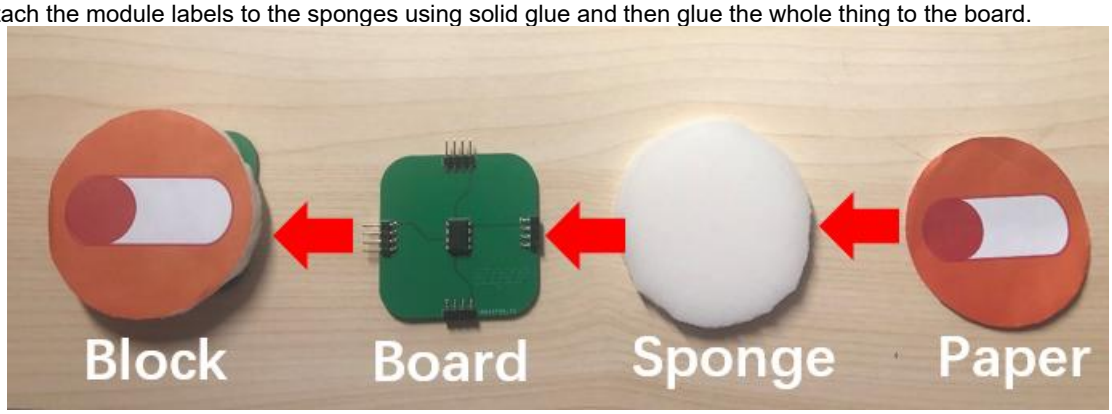

Figure 3: The concatenation of a block of the tangible system

The system consists of modules with different instructions. The specific operations performed by each module correspond to the sponge tag on its block. For instance, the traditional parameters are encapsulated in a knob control block, which can adjust various values such as speed, brightness. Moreover, the branch block means it can do multiple things at the same time.

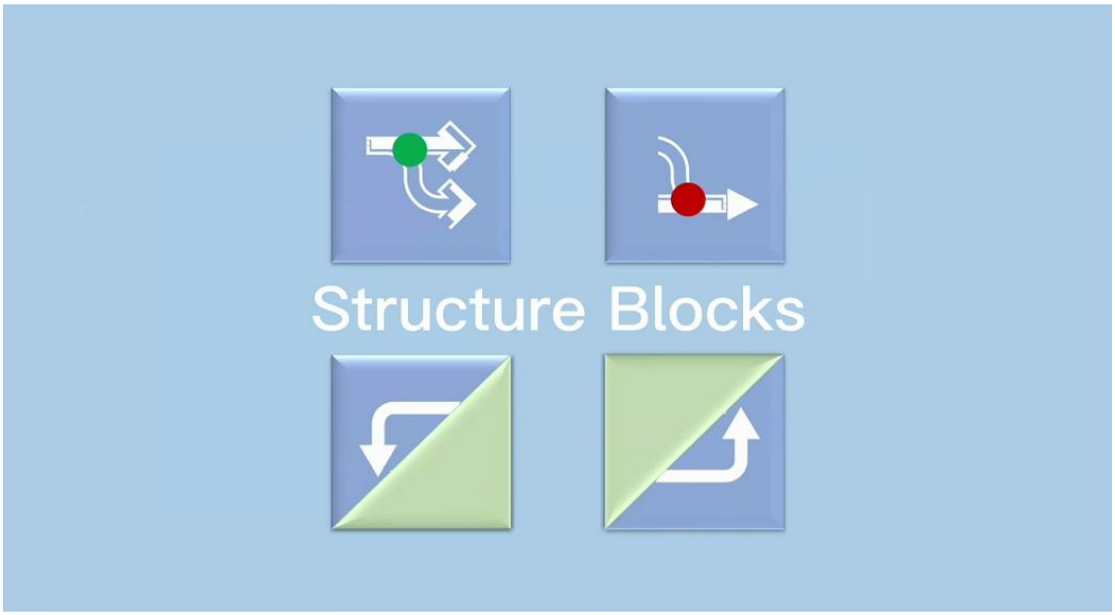

Figure 4: The Structure Blocks

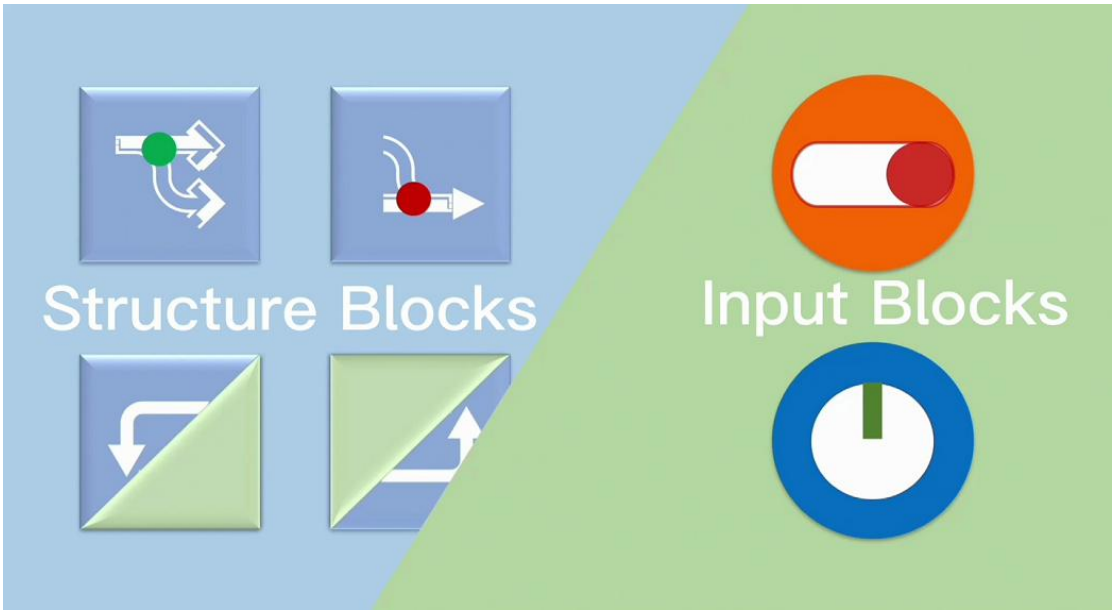

Figure 5: The Structure Blocks and Input Blocks

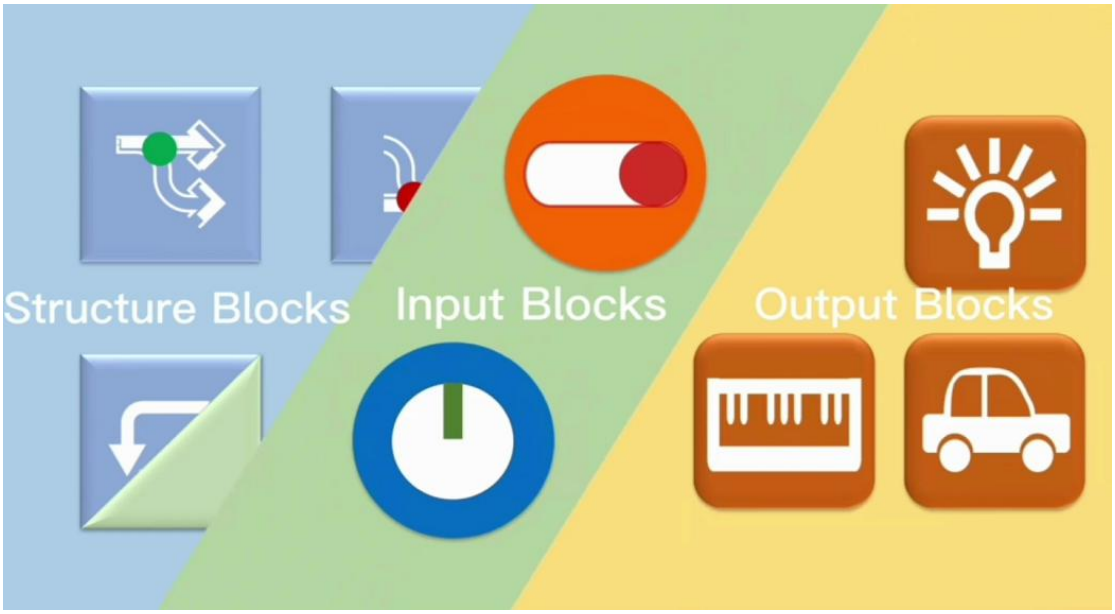

Figure 6: The Structure Blocks, Input Blocks and Output Blocks

And there are two ways of connecting modules: connectors or wires.

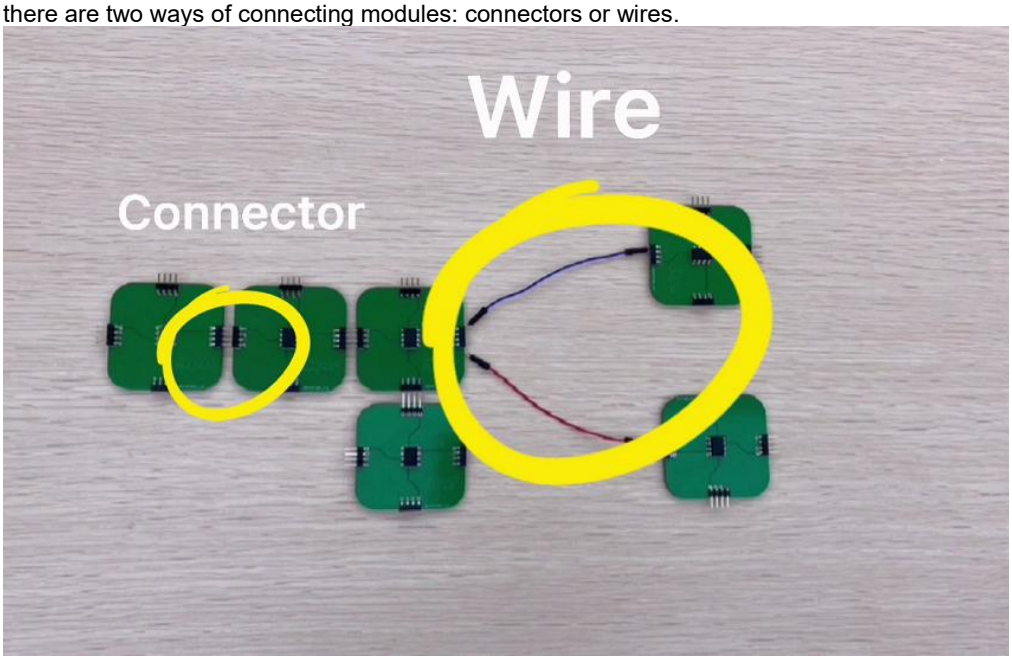

Figure 7: Two ways of connecting modules

The connection methods can be chosen according to users' requirements. In addition, the code in each module has been built in by the developer, so users can assemble the blocks according to their needs. Once

assembled, users can connect to the corresponding actuators via USB cable. Then, the physical toolkit delivers the commands to them.

The hallmark of our tangible programming language modules is that users can rearrange the physical modules according to their needs to achieve different kinds of purposes.

Three specific cases are presented below (Green means the start and the red means the end):

1. Control the brightness of the light. Users can use loops, switches, bulbs and other physical codes spliced together to achieve controlling the light.
2. Output the custom music. Users can input created scores into the music module through an external device by connecting the music block to the instrument block. Then customized music can be output.
3. Control the multifunctional car. Users can use the loop and branch blocks to combine the two functions above. Then, let the car make a sound and travel on command while the lights stay on at the same time.

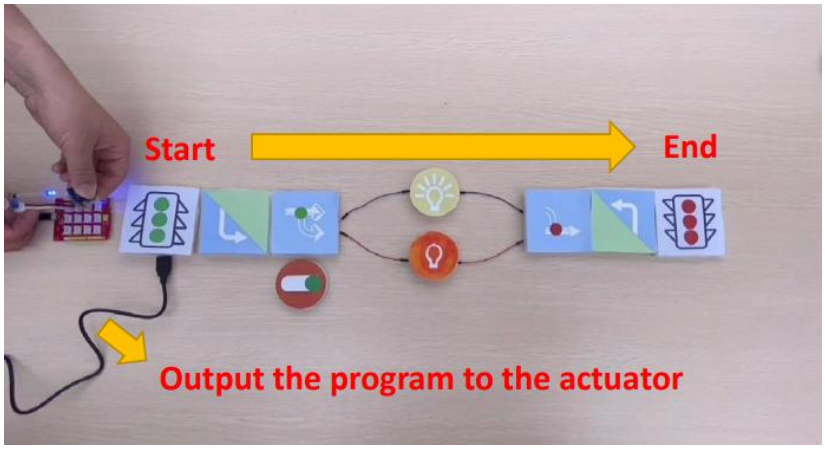

Figure 8: Control the brightness of the light

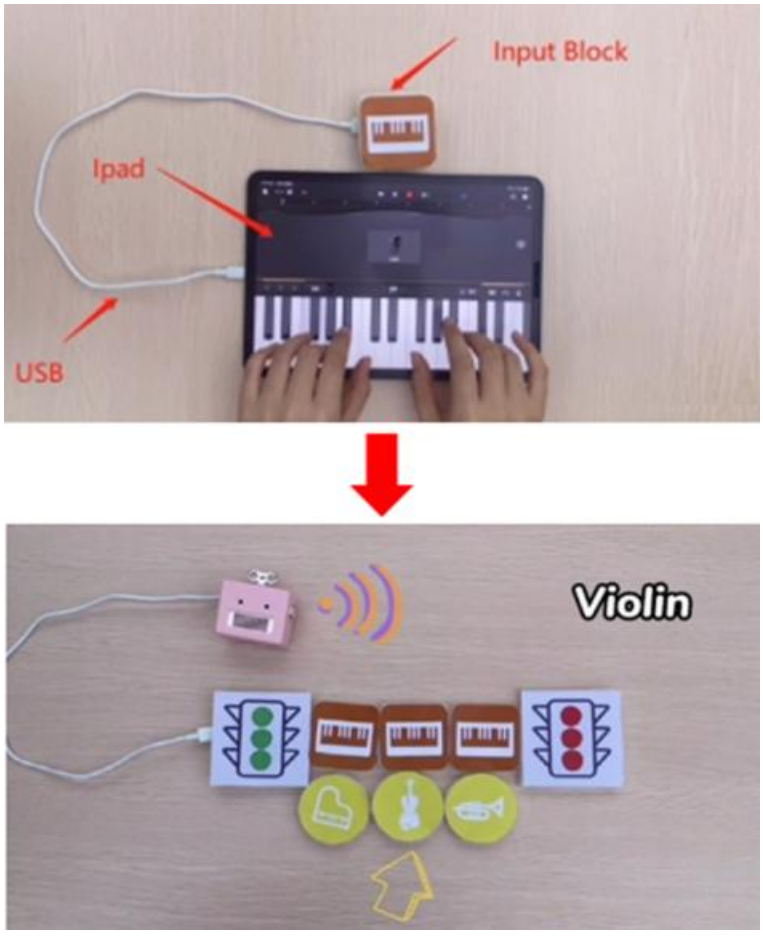

Figure 9: Output the custom music

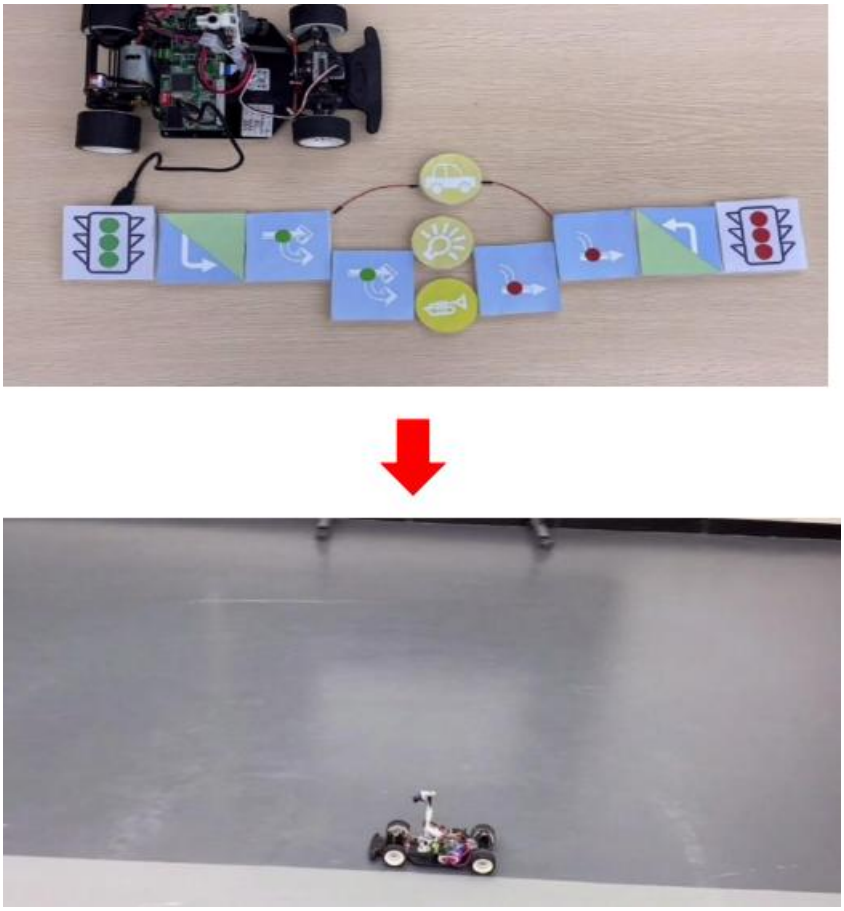

Figure 10: Control the multifunctional car

4 EVALUATION

Several studies have been conducted to evaluate the usability of our products. These studies involved 14 children with same age from Nanhai Middle School. They programmed using modules in a laboratory environment.

In the latest study, the potential for developing computational thinking in children's programming activities were concerned.

4.1 Evaluation points

In this study, several research questions are quite essential:

1. Do kids like programming with the modules?
2. Is the programming module suitable for children?
3. What are they interested in?

Hypothesis:

- H_0: Different genders do not affect the satisfaction of the product
- IV: The gender of the 14 participated children
- DV: The satisfaction rate of these 14 children

4.2 The evaluation process

There are two main phases in the evaluation research: task completion (implement the bulb lighting program) and task creation (Design a program that can achieve more functions and draw representation).

In the task completion section, children were asked to complete two different levels of tasks.

During the task creation process, the children had no restrictions, they could use the same building blocks to create new model drawings and there was no time limit in both tests. In order to accomplish these tasks, the children must design a feasible solution based on a given design drawing, which is constructed by using a custom programming block builder.

The assembly of modules is a kind of simple task-driven creation process. Simple tasks only require the realization of the most basic functions. And complicated game not only to achieve the most basic function, also includes some additional features, the children need to use a certain number of modules to build the final prototype, it has more than one kind of feasible solution, the amount and type of modules can be based on personal eventually want to reach the effect and function of the quantity to determine.

4.3 Satisfaction evaluation

It is important to know what the kids think of the tool, their programming experience, or whatever it is that sticks in their mind.

Fourteen children, 7 girls and 7 boys, took part in the study. They all had some experience with computers, but none of them knew how to program or use any programming tools.

Therefore, a questionnaire has been given out and based on this, the data satisfaction of the product is calculated.

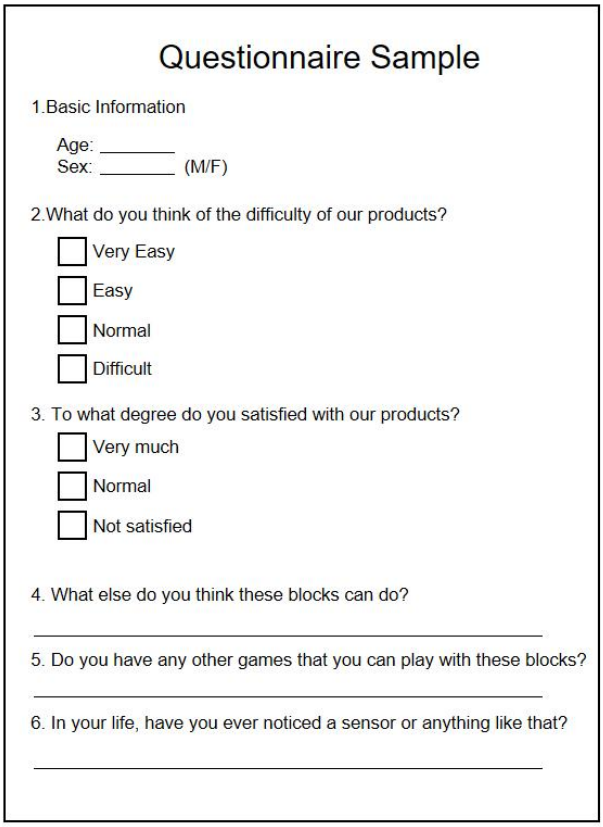

Figure 11: Questionnaire Sample

Table 1: Notation

Notation	Description
N_A	Size of the sample who are very satisfied with the product
N_B	Size of the sample who feel the product is normal
N_C	Size of the sample who are not satisfied with the product
N	Size of the sample
σ	Grand Standard Deviation
σ_1	Standard deviation of data for Group 1
σ_2	Standard deviation of data for Group 2
R	Satisfaction rate
$\overline{R_1}$	Mean of data for Group 1
$\overline{R_2}$	Mean of data for Group 2

User Satisfaction Survey Results and Comprehensive Satisfaction Calculation:

$$R = \frac{N_A \times 100\% + N_B \times 70\% + N_C \times 50\%}{N}$$

Table 2: Data set and Satisfaction rate

User	Product satisfaction (Group 1) Male	Product satisfaction (Group 2) Female
1	A	B
2	A	A
3	B	A
4	A	A
5	B	B
6	A	B
7	A	A
R	91.43%	87.14%

$$\sigma_i = \sqrt{\frac{\sum_{i=1}^{N}(N_i - \mathrm{R})^2}{N}}$$

$$\sigma = \sqrt{\frac{{\sigma_1}^2 + {\sigma_2}^2}{2}}$$

Table 3: The Standard Deviation, Grand Standard Deviation and Mean

SD	Decimalize	Mean	Decimalize
σ_1	0.04495297	$\overline{R_1}$	0.9143
σ_2	0.07743656	$\overline{R_2}$	0.8714
σ	0.01763254	R	0.8929

$$t = \frac{\overline{R_1} - \overline{R_2}}{\sigma \times \sqrt{\frac{2}{N}}}$$

$$d = 2N - 2$$

Table 4: The P-value, degree of freedom, users number, GSD and T-value

P - value	0.01
Degree of freedom	12
Users	7
Grand Standard Deviation	0.01763254
T - value	4.55177243

Table 5: Critical values for a two tailed test. (part of the whole table)

Degrees of Freedom	p-value=0.05	**p-value=0.01**	p-value=0.001
11	2.201	3.106	4.437
12	2.179	**3.055**	4.318
13	2.16	3.012	4.221

When d = 12 and p-value = 0.01, T - value = 4.552 > 3.055.

So it can prove the hypothesis{¥ H}_0: Different genders do not affect the satisfaction of the product

4.4 Results and discussion

A comprehensive analysis of observation notes, audio recordings, and interview transcripts has been conducted to find answers to research questions from the analysis and statistical analysis of children's speech acts.

To know whether children could create their own computer programs is also important in this research.

The number of modules and the amount of time the children spent on simple and complex programming tasks were similar. In addition, they are even more involved in such aspects as module design and creation, and are happy to spend more time creating and exploring.

Table 6: The conclusion of three main aspects of evaluation

Effectiveness	Children can make the things that they want by assembling block modules.
Efficiency	For simple tasks, the average time they need is 3.68 min and 7.14 min for complex tasks. Which means that children have ability to complete any task in a very reasonable time
Satisfaction	Children are satisfied with our product because the data in following tables show that they spent more time on creating things and that indicated the children enjoy creating by using our blocks

Table 7: The time(unit: minute) and blocks children used in the study

	Task complete session				Task creation session	
Number	Simple		Complex			
	Time(min)	Block	Time(min)	Blocks	Time(min)	Blocks
1	3.50	9.00	7.00	18.00	9.00	18.00
2	3.00	9.00	5.00	19.00	10.00	26.00
3	4.50	9.00	7.00	22.00	13.00	18.00
4	4.50	8.00	9.00	21.00	8.00	16.00
5	3.00	9.00	8.00	20.00	11.00	31.00
6	4.50	9.00	7.00	21.00	10.00	27.00
7	4.00	8.00	9.00	18.00	9.00	16.00
8	3.00	9.00	5.00	20.00	8.00	25.00
9	4.50	8.00	7.00	19.00	12.00	17.00
10	3.00	9.00	8.00	22.00	11.00	24.00
11	3.50	8.00	9.00	18.00	9.00	26.00
12	4.00	9.00	5.00	19.00	10.00	19.00
13	3.50	9.00	7.00	18.00	8.00	23.00
14	3.00	8.00	7.00	20.00	9.00	18.00
Avg	3.68	8.64	7.14	19.64	9.79	21.71

Eight chose "Very Easy", five chose "Easy", and one child chose "Normal". In addition, the children showed very excited emotions in the module design and creation, showing the most abundant behaviors.

From this it can be concluded that Tangible Program Block is very suitable for children.

4.5 Conclusion

Although the study was relatively small, from the results of the study, it showed that Tangible Program Block has some computing concepts in its programming activities, and children really handle them well.

This study further shows that Tangible Program Block is easy to use for children, who are able to create and build their own programs while playing.The point is that Tangible Program Block have the potential to help raise children's awareness of computer programming concepts.

Their boundless imagination has inspired the original team(Team2) to modify and upgrade our products in the future

5 CONCLUSIONS

The improvement of programming ability and logic is becoming more and more important in education. Also, the openness of the product can attract more children to join the learning of programming. Thus, the Tangible Program Block composed of circuit boards and buttons can be used for children to program, which will satisfy their curiosity. This article first introduces the logical design of the tangible programming language, the composition and function of the physical modules. After that, three examples of lights, music, and cars were shown, which fully proved the openness and scalability of the product. Unlike other tangible programming products in the market, our products have limited encapsulation of C language, allowing children to realize more functions without understanding specific codes.

In addition, the product has a broad circumstance to be used, it can be operated for not only program learning, but also daily entertainment. Although it is designed for children, people at any age can have a try. Besides, according to the result of evaluation, users can try to design complex blocks combinations to achieve the combined function of the actuator and increase the challenge. Additionally, the product has no requirement for the place, users can operate anywhere. When the physical block and circuit are assembled, users can use USB to transfer the generated instructions to the designated device. Then, the devices will execute the corresponding program. In addition, we have obtained a good evaluation through a survey of 14 children. In the future, we will continue to work hard to continuously improve products and develop more functional blocks for users to explore and make more challenged combinations, so as to stimulate children's interest in programming and better develop their programming logic.

REFERENCES

[1] Michael S. Horn and Robert J.K. Jacob. 2007. Designing tangible programming languages for classroom use. Retrieved May 19, 2021 from https://www.researchgate.net/publication/221308672_Designing_tangible_programming_languages_for_classroom_use.

[2] William J. Rapaport. 2004. Great Idea III: The Boehm-Jacopini Theorem and Structured Programming. Retrieved May 17, 2021 from https://cse.buffalo.edu/~rapaport/111F04/greatidea3.html.

[3] Edsger W. Dijkstra. 1968. Go To Statement Considered Harmful (Letter to the Editor). Communications of the ACM 11(3) (March): 147-148.

[4] Fiona Harwood. 2018. Wizard of Oz testing – a method of testing a system that does not yet exist. Retrieved May 18, 2021 from https://www.simpleusability.com/inspiration/2018/08/wizard-of-oz-testing-a-method-of-testing-a-system-that-does-not-yet-exist.