プログラミング学習支援のための ビジュアル言語デバイスの開発

CAMPAÑA ROJAS JOSÉ MARÍA[†1]　　小林 孝浩[†1]
平林 真実[†1]　　鈴木 宣也[†1]

概要：プログラミングの苦手なデザイナーに向け、プログラミングの理解を支援する方法を探る研究である。変数と条件文の 2 つのプログラムの基本要素の理解を目標とした。まず最初にデザイナーとデザイナーを志す学生に調査を実施した。調査から、プログラミングの要素を実際のブロックに置き換えたデバイスを開発し用いることとした。ブロックを並べ接続すると、ビジュアルな物理的要素と実際のプログラミング言語の間の関係をディスプレイ上に表示することができる。

Development of a Physical Device That Represents a Visual Language to Aid in Learning a Programing Language.

JOSÉ MARÍA CAMPAÑA ROJAS[†1]　TAKAHIRO KOBAYASHI[†1]
MASAMI HIRABAYASHI[†1]　NOBUYA SUZUKI[†1]

Abstract: This research seeks for a solution that allows designers who lack programming skills to understand that process. It focuses on the comprehension two of basic parts of a program: variables and conditionals. It was first approached by taking surveys to advanced design students and professionals. Then the development a device that uses physical blocks to represent parts of a program. When the blocks are connected, the real can be seen on a display to show the relation between the physical visual components and the real programming language.

1. Introduction

This research tries to find a solution to a problem that is occurring in the job of the designers: the discipline of Design has became interdisciplinary, specially in relation to information technologies, requiring coding skills that designers do not possess. Almost every designer whether for a project in classroom or for work has the necessity to write code at some point. This research tries to seek for a solution that allows designers to understand the process of writing code focusing on the comprehension of some basic parts that conform a program, for example, variables or conditionals and make them able to actually use them.

2. Background

Research is being done to teach information technology students at university level [1] using for example, social games [2] to motivate the students and other systems. And there is also is a lot of research being done to develop systems to teach children to manipulate information technologies. But there are just a few that are directed to teach designers or adults besides university classes, tutorials or video tutorials. In many cases visual languages are being applied in this case to teach children, but it is questioned if they help in to the comprehension of coding, that is because they replace the need to write it by using space and visual actions. Using current tools to learn programming such as Scratch [3] [4], it is pretty straightforward to create a simple program, but almost all of these systems do no show the code. An exception is made for Google Blockly [5], which has the ability to output the source code. The ability for the user to be able to see the code for a program that was created in a very easy way gives them the plus to make them able to study it and apply it later. Designers very often end copy-pasting code from some website when they want to write a specific program. In the case of novice designers that do not comprehend the code, it makes them almost impossible to modify it to suit his or her needs. With a little background in computer programming the story changes by much.

3. Approach

To approach this problem a 22-question survey was applied to 88 advanced design students and design professionals in Japan and Mexico. Interviews where also applied to some designers. According to the survey, designers have knowledge of HTML and CSS, which are two of the current tools used to create

†1 情報科学芸術大学院大学
Institute of Advanced Media Arts and Sciences

websites. Designers have the necessity to program logic for their web works and even convert them in to web applications or application for devices. All this can be achieved with HTML, CSS and JavaScript using a framework like Apache Cordova [6]. That is the main reason why this research focuses on teaching the basic principles of JavaScript to designers (which are basically the same for any programming language).

There is extensive research [a] that supports that, when teaching abstract concepts to kids, it is better if those abstract concepts are represented in a tangible way. This is one of the main ideas of the Montessori Method, which can also be applied to adults [b]. First, there was an experiment with cards that represented different parts of JavaScript as in a visual programming language. These cards were evaluated with the user but since they don't give real time feedback, they didn’t serve much to teach the concepts. There was also the problem that the user does not know what is the way to connect them or to evaluate if it has been done right. After that a test was made to find if what is the best way to transfer this knowledge: using paper cards, using just an iPad application or using a device that merges an application and physical components. It turns out that just an application is the easiest to use, but using physical components make a better connection to the user. So after this experiment the development of a first iPad application / physical device hybrid prototype was started. This prototype was later called js.bit. js for JavaScript and bit as in little piece and the information bit used in computers.

4. js.bit

js.bit is a set of 28 blocks (or bits) and an iPad application. Each block represents some part of a program, for example, a variable. When the blocks are connected with each other in a certain pattern, a program is created, just like in a visual language. The iPad application interface is divided in to two spaces: block space and source code space. When the blocks are placed (and connected) over the block space, a block appears in the block space, and the corresponding code appears in the source code space. A line (colored depending on the type of block) then gets from the block to the corresponding line in the source code indicating the correct grammar and how the block would act if it was real JavaScript. So the source code that was created with the blocks can be seen, read, analyzed and executed. The code can also be copied (it can’t be copy-pasted) to a text editor and execute it like normal JavaScript and it will run (if the proper arrangements for the output blocks is done). The blocks are divided in to two categories, logic blocks and output blocks. One set is composed of six value blocks, three variable blocks, four operator blocks, two conditional start blocks, two else blocks, two conditional end blocks, one loop start and one loop end block, one free input block, two RGB light block (lightbox), two servomotor block (motorbox) and two numeric display block (logbox) and the main block.

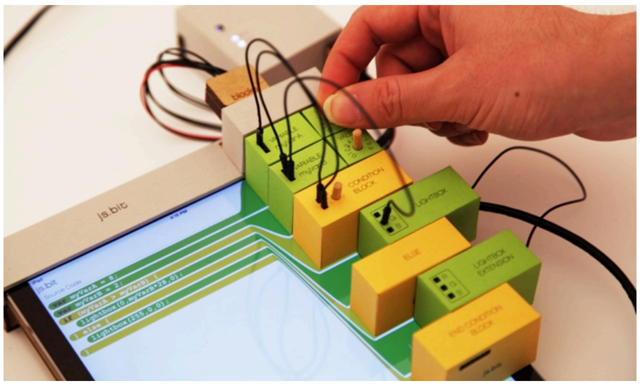

Fig. 1 js.bit

4.1 js.bit system specifications

Every block has 6-pin connector on their front and in their back. This connector is used to give power to the blocks and to enable them to exchange information between them. Also every block has a microprocessor on it’s inside to communicate to the system their status. Depending on their function the microprocessor is either an ATMEGA ATTINY 85 or an ATMEGA 328. The blocks communicate to each other using the I^2C protocol. Some blocks have input or output connector on the top or on the right side. The main block acts as a relay that passes messages to other blocks or to the iPad application via Bluetooth and also detects the order of the array of blocks using a relay algorithm.

4.2 Block user testing

As of the time of writing this manuscript, the js.bit system is currently being tested with designers. The tested users are advanced students of design or professional designer. The test consists in an interview, activities that involve the creation of several program using the blocks, another part of the test is done in the computer, asking the subjects to write the same exercises made with the blocks but this time in a text editor using real JavaScript. At the end of the test a survey is also taken. This survey contains some questions that serve as a clue as if the user understood the concepts or not.

a) For example, research made by Jean Piaget or Maria Montessori.
b) This research is borrowing some concepts of the Montessori teaching system but its intention is not to create Montessori teaching materials.

Fig. 2 js.bit user testing

The programs that the user are asked to make involve the use of numbers, variables and conditionals. For example writing a value in the logbox (a block that has a numerical display), or depending on the value of two variables turn red or green the lightbox (a block that can be lit with any color).

4.3 Description of blocks used in the user testing

For the user test mainly five logic blocks are used: value block, variable block, conditional start block and conditional end block. The value block consists of a knob that is pointing to a number that ranges from 0 to 9. When this knob is turned right or left, the number can be changed and hence the value that it represent. It connects to other blocks using a 3-pin connector.

The variable block has a 6-pin connector on its front and back, in its right side has a 3-pin connector that receives a value block and in its top, has 3 female pin connector that serve as input and out put of its value. It connects to other blocks using a cord (jumper wire). These blocks require a connection to the variable in order to be able to take its value.

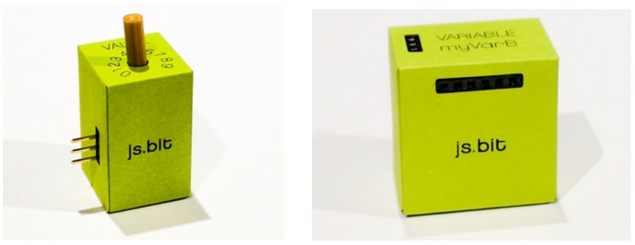

Fig. 3 left: value block, right: variable block

The conditional blocks consist of three blocks; conditional start block, end condition block and else block. They represent an if statement. The condition start block has a knob and two pin inputs. For the block to work its inputs need to be connected to other corresponding blocks in order to create a condition, for example, variable blocks. When the knob is turned, the condition can be changed, its ranges are: == (equal), != (not equal), > (greater than), < (minor than), >= (great or equal than) and <= (minor or equal than). The else block is used to indicate what to do when de condition is not met; the use of this block is optional. The end condition block is used to delimit the conditional. The else block and end conditional block does not require or give any physical input or output.

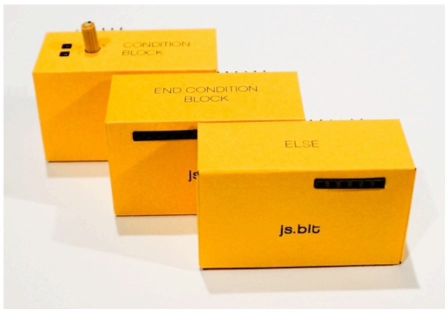

Fig. 4 js.bit conditional blocks

5. Conclusion

The discipline of design is requiring more than ever the necessity of the designers to be able to work with information technologies. It is necessary to find a way to push designers programming knowledge, and adult people in general. As of this time, js.bit is showing some favorable results in the user testing with designers but the testing is not over yet. As it uses a tangible object, and has an appealing design, it is easy to connect to the user, and because the user can have a physical response it makes it fun. And the fact that is fun makes it able to give a push to the level of understanding of JavaScript of the designer.

References

1) Tony Jenkins, Teaching programming – A Journey from Teacher to Motivator. Proceedings of 2nd Annual LTSN-ICS Conference, pp. 67-51, 2001.
2) Ben Leong, Zi Han Koh, Ali Razeen. Teaching Introductory Programming as an Online Game. Available at:http://www.comp.nus.edu.sg/~bleong/teaching/teaching-online-game.pdf [Accessed December, 2014].
3) MIT Scratch. http://scratch.mit.edu
4) Maloney, J., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. Programming by Choice: Urban Youth Learning Programming with Scratch. SIGCSE conference, Portland, pp.367-371, 2008.
5) Google Blockly https://code.google.com/p/blockly/
6) Apache Cordova http://cordova.apache.org