Abstract

With the growing interest in computational thinking (CT) and coding education worldwide, various discussions have been raised that there is a need to practice the coding education to K-12 especially from an elementary school. Accordingly, several block-based educational programming languages have been developed such as Scratch, App Inventor, code.org and so on for introducing coding education, and various related experimental studies have been carried out. 

These block-based programming languages are very useful and convenient, and it has the advantage of being able to help students understand the logic and control structure at the initial programming stage. However, if the blocks increased, there are such inconveniences that readability becomes worse, and therefore the debugging process can be more difficult. In addition, the block-based programming languages are confined in a particular web page or system, so there is shortcoming in the transition to the practical programs.

In this study, to address these concerns, we suggest a text-based introductory educational programming language, that is, JQuery Mobile based on the mobile web browser on smart devices, and proposed the benefits and differences compared to the block-based languages in instructional class situations.

Keywords: Coding Education, JQuery Mobile, Educational Programming Language.

1 INTRODUCTION

Today is the era of software and computing. With growing worldwide interest in computational thinking (CT) and coding education, arguments have been raised that there is a need to apply the coding education from an early age. Countries are eager to train students and workers with creative and computational thinking to gain a competitive advantage in the global arena and the 21st century society [1]. Therefore, they are putting a lot of effort into integrating computing abilities into the educational environment. In the U.S. and U.K., curricula and training programs have been devised to help primary and secondary school students acquire computing skills [2][3][4][5]. In Korea, students gain computer-programming skills via a variety of educational programs provided in the public and private sectors [6][7][8][9].

In this regard, several block-based educational programming languages have been developed such as Scratch, App Inventor, code.org and so on for introducing coding education, and various related experimental studies have been carried out. These block-based programming languages are very useful and easy to use, and they have the advantage of helping students to understand logic and control structures in the initial programming stage. However, as the number of blocks increases, there are such inconveniences that readability becomes worse. In this case, the debugging process can be difficult. In addition, block-based programming languages are confined to a particular web page or system, so there is a shortcoming in the transition to practical programs.

In this study, to address these concerns, we suggest a text-based introductory educational programming language called JQuery Mobile, which utilizes a mobile web browser on smart devices, and discuss the benefits and differences of JQuery Mobile in relation to block-based languages in instructional class situations.

2 THE STUDY

2.1 Coding Education

The CSTA Standard Task Force (2011) identified five strands of computer science education. It is
noteworthy that CT and computing practice are different strands. CT encompasses problem-solving methodology with algorithmic procedures and computing [10][11], and computing practice refers to the technical aspects associated with the actual programming practice [4][5]. Coding education is associated more with both the former and the latter (See Fig. 1).

![Fig. 1 5-strands and 3-levels of the computer science education (CSTA, 2011)](image)

The importance of mathematical/scientific thinking abilities has long been recognized, and computing skills occupy an equally important place in information-oriented society today [1]. Given the emphasis on computing, arguments have been put forward for providing coding education from an early age to gain a competitive advantage in the global arena and the 21st century. Accordingly, as noted in the introduction, the U.S. and the U.K. have developed curricula and training programs to help primary and secondary school students acquire computing skills [2][3][4][5]. In Korea, students gain computer-programming skills via a variety of educational programs provided in the public and private sectors [6][7][8][9]. Fig. 2 shows two famous programs of coding education in Korea.

![Fig. 2 Junior Software Academy (Samsung) and Campaign of playing with Software (Naver) websites](image)

### 2.2 Block-based educational programming languages

The ACM (2003) and Pepper (2007) argued that educational programming language is necessary to increase algorithmic thinking and to reduce difficulties in programming languages per se, as well as to alleviate the fear of coding and debugging.

Scratch and App Inventor, educational programming languages developed by the MIT Media Lab, are widely used in educational fields. The productions of the Scratch are animation and game programs based on the Adobe Flash player, and the results of the App Inventor are application package (apk) files executable on Android-based smartphones (See Fig. 3). Diverse instructional contents utilizing Scratch/App Inventor programming has been developed. Studies of the effects of Scratch and App
Inventor on learning flow and programming abilities, programming learning attitudes, self-directed learning abilities, learning motivation and academic achievements, logical thinking, and learning style and problem-solving ability and so on have been conducted.

Fig. 3 Scratch and App Inventor

Code.org is a non-profit organization and eponymous website led by brothers Hadi and Ali Partovi that aims to encourage people, particularly school students in the United States, to learn computer programming. The website includes free coding lessons. Code.org also targets schools in an attempt to encourage them to include more computer programming classes in the curriculum (See Fig. 4).

Fig. 4 Hour of Code lesson and Teacher Dashboard of the Code.org

2.3 jQuery Mobile

jQuery Mobile is a cross-platform framework for creating mobile web applications (web apps) accessible on all smartphone, tablet as well as desktop devices. It is built on top of jQuery, a JavaScript library designed to simplify client-side scripting, and therefore it has a minimal learning curve for students already familiar with jQuery syntax. jQuery Mobile simplifies and enhances the development of mobile web apps by integrating HTML5, CSS3, jQuery and jQuery UI into one framework. The framework is compatible with all major mobile platforms as well as all major desktop browsers [12][13].

A screen of the project is defined by a <section> HTML5 element, and data-role of page. Data-role is a jQuery Mobile construct, and not an HTML5 one. A page may have header and footer elements with data-role of header and footer, respectively. In between, there may be an <article> element, with data-role of content. Lastly, a <nav> element, with data-role of navbar may be present.

One HTML document can contain more than one <section> element, and thus more than one screenful of content. This way it is only necessary to load one file which includes multiple pages of content. One page can link to another page within the same file by referencing the page's id attribute (e.g. href="#second"). Following is the brief explanation of the data attributes used in example codes:
• **data-role** – Specifies the role of the element, such as header, content, footer, etc.
• **data-theme** – Specifies which design theme to use for elements within a container. Can be set to: a or b.
• **data-position** – Specifies whether the element should be fixed, in which case it will render at the top (for header) or bottom (for footer).
• **data-transition** – Specifies one of ten built-in animations to use when loading new pages.
• **data-icon** – Specifies one of fifty built-in icons that can be added to an element.

![Sample codes and the result of jQuery Mobile](image)

Fig. 5 Sample codes and the result of jQuery Mobile (From Wikipedia) [14]

### 3 METHODOLOGY

In this study, we compared the efficiency and the conditions of two educational programming courses that used block-based tools and jQuery Mobile. They were both six-week courses that teach Scratch/App Inventor and jQuery Mobile programming, and it took two hours per week over the period of the courses. The students took the jQuery Mobile programming class from May 2013 to June 2013 and the Scratch/App Inventor programming class from June 2014 to July 2014. Eighty-four IT-gifted students in the fifth and sixth year of primary school took part in the experimental courses, with 34 students in the jQuery Mobile group and 50 students in the Scratch/App Inventor group. Both courses adopted the framework of creative problem solving [15], and they were administered via online classes with missions and VOD lectures. Table 1 shows the course plan of each programming lesson.

<table>
<thead>
<tr>
<th>Weeks</th>
<th>jQuery Mobile class</th>
<th>Scratch class</th>
<th>App Inventor class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Basic instruction to use the editor (Editplus3) and the FTP application. What is a web app?</td>
<td>Basic instruction to use Scratch. What is Scratch?</td>
<td>Basic instruction to use App Inventor. What is App Inventor?</td>
</tr>
<tr>
<td></td>
<td>Creating a basic page template with jQuery Mobile</td>
<td>Creating an advanced program with Scratch (Using Sensing, Control, Operators, Data, etc.)</td>
<td>Creating an advanced program with App Inventor</td>
</tr>
<tr>
<td>2</td>
<td>Learning about the UI components of jQuery Mobile to format contents and create lists, navigation bars, and buttons</td>
<td>Creating a basic program with Scratch (Using Motion, Events, Looks, Sound, Pen blocks)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Finding the topics or problems for programming solutions (from week 1~ week 3)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Course plans of the each programming lesson.
The students were motivated to be self-directed throughout the entire course. Each student chose a problem to work on alone and created the result (i.e., a web app, a Scratch program or an Android-based application) according to his or her own design. In the first 3 weeks, they learned the basics of language and identified topics or problems for programming solutions. In the 4th week, the students collected and reviewed various supporting material to perform the abstractions of the problem and represented these in a structured tree. They then created a storyboard for each element of the tree. During this period, they devised frames and shapes for the abstract ideas they had developed in the first stage of the process. In the 5–6th week, the students implemented a web app, Scratch program, or application based on their design. In this stage, the students translated their ideas into the codes of the respective programming language.

After completing the 6-week course, the students provided feedback online and gathered in the same offline place to share their outcomes and feedback.

### 4 RESULTS

The ultimate goal of our programming course was to cultivate problem-solving abilities in the students, as well as to advance their programming knowledge. With this aim in mind, the students used different tools to accomplish the same goal. After our experimental course, we found some similarities and differences between the block-based tools and jQuery Mobile, as shown in Table 2. Figures 6 and 7 depict the outcomes of our experimental course.

<table>
<thead>
<tr>
<th>Similarities (Common benefits)</th>
<th>Scratch / App Inventor</th>
<th>jQuery Mobile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to use for students new to programming</td>
<td></td>
<td>Text-based web programming language</td>
</tr>
<tr>
<td>Allows users to utilize event-driven programming</td>
<td></td>
<td>Utilizes pre-built widgets and HTML elements (easy to use)</td>
</tr>
<tr>
<td>Produces interactive outcomes</td>
<td></td>
<td>Deals with the user-interface layouts.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Differences (advantages and disadvantages)</th>
<th>Scratch / App Inventor</th>
<th>jQuery Mobile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block-based animation/Android application programming language</td>
<td></td>
<td>Text-based web programming language</td>
</tr>
<tr>
<td>Utilizes the objects and components</td>
<td></td>
<td>Utilizes pre-built widgets and HTML elements (easy to use)</td>
</tr>
<tr>
<td>Deals with the logic/control structure</td>
<td></td>
<td>Deals with the user-interface layouts.</td>
</tr>
<tr>
<td>Requires online website registration</td>
<td></td>
<td>Requires an editing program (Notepad, Editplus3, Coda2 for Mac, etc.)</td>
</tr>
<tr>
<td>Provides cloud storage</td>
<td></td>
<td>Needs server storages for access via a web service</td>
</tr>
<tr>
<td>Difficulties finding errors in complex programs</td>
<td></td>
<td>Easy to find errors in complex programs using developer tools in web browsers</td>
</tr>
<tr>
<td>Runs on the specific platform (website on a PC, smartphone/virtual emulator)</td>
<td></td>
<td>Runs on HTML5-supported web browsers of PCs, smartphones and tablets (Chrome, Safari, Opera, Firefox, IE11, etc.)</td>
</tr>
</tbody>
</table>
After completing the 6-week courses, feedback from the participated students was gathered online. Common expressions found in their comments included “making my own decisions,” “achievement,” “confidence,” and “more interested in the subject I selected.” In addition, some students wrote that the experience of developing a web app or an App Inventor program that they could run on their own smartphones motivated them to consider becoming a professional software developer.

5 CONCLUSIONS

Dijkstra (1971) pointed that programming is a methodology of constructive reasoning applicable to any problem of algorithmic solution [16]. Thus, to strengthen this programming capability, a variety of instructional methods and tools should be utilized. The objective of our experimental programming course was to magnify the power of human thinking with the capabilities of CT and the computer using a variety of tools and methods.

Block-based programming languages are very useful and convenient, and it has the advantage of being able to help students understand the logic and control structure at the initial programming stage. But block-based programming languages are confined in a particular web page or system, so there is shortcoming in the transition to the practical programs. This study shows that jQuery Mobile is another effective introductory tool to teach young students basic programming skills with experiences of practical programming environment, i.e., text-based environment.
It would be useful to study other types of classroom practices that exploit jQuery Mobile to strengthen students’ CT capabilities and programming knowledge. To fully realize the objectives of coding education, further studies are needed on how best to design coding education classes and utilize various tools appropriately and efficiently.

6 ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2014-S1A5A2A01013263)

REFERENCES