Verification of the effectiveness of classes using intermediate content for transitioning from a visual- to a text-based programming language

Katsuyuki Umezawa
Dept. of Informatics
Shonan Institute of Technology
Kanagawa, Japan
umezawa@info.shonan-it.ac.jp

Makoto Nakazawa
Dept. of Industrial Information Sci.
Junior College of Aizu
Fukushima, Japan
nakazawa@jc.u-aizu.ac.jp

Shigeichi Hirasawa
Research Institute for Sci. and Eng.
Waseda University
Tokyo, Japan
hira@waseda.jp

***Abstract*—In recent years, visual programming languages (VPLs), such as Scratch, have been widely used by programming beginners. Subsequently, learners often transition to text-based programming languages (TPLs), such as Java. However, a seamless transition between these two types of programming languages has not yet been effectively established. In this study, we experimentally demonstrated that students who utilized our proposed intermediate music content between learning VPL and TPL exhibited enhanced comprehension of the TPL. Our results indicate that students who engaged with intermediate content while transitioning from VPL to TPL during their high school years performed better in their first-year college programming classes compared to those who did not use such content.**

***Index Terms*—programming languages, visual-based languages, text-based languages, learning analysis**

I. Introduction

We initiated a research project aimed at establishing a methodology for transitioning from a visual programming language (VPL) to a text-based programming language (TPL). This study focuses on examining and prototyping educational content that leverages the advantages of both VPL and TPL learning to bridge the gap between the two. The evaluation was conducted through empirical experiments. This report forms part of the project and investigates whether high school students who took classes using our proposed intermediate content showed improvement in their grades in university programming-related classes.

II. Previous work

Several studies have been conducted on the relationship between VPLs and TPLs. Robinson [1] investigated the transition from a VPL, such as Scratch, to a TPL. Tóth et al. [2] highlighted the existence of a gap between VPL and TPL learning.

Our aim is to establish a seamless transition method and experimentally demonstrate that learning with our proposed intermediate content enhances comprehension of TPL [3]. Additionally, we assessed our proposed intermediate content through a questionnaire and found that it possesses intermediate features between VPL and TPL [4]. The biometric information of learners were evaluated during the VPL and TPL sessions [5] [6].

III. What is intermediate content?

A. Hypothesis

The proposed intermediate content should be simple, requiring no additional knowledge specific to TPL, provide quick feedback (immediate results), be resistant to grammatical errors, and facilitate the easy location of logical errors.

B. Realization

As content embodying the aforementioned characteristics, we developed educational material for creating music in JavaScript using JSFiddle [7]. As illustrated in Fig. 1, JSFiddle is a web-based integrated development environment that allows users to code in JavaScript and view execution results on a single screen via a web browser. By incorporating a library (Beeplay) into JSFiddle, music creation becomes straightforward. Although JSFiddle is not a visual-based language, we anticipated that content developed with the previously outlined features would yield the positive effects discussed in subsequent sections.

C. Expected effects

The use of the "play" method eliminates the need for memorizing many reserved words or grammar rules, enabling immediate sound production and swift verification of results. Identifying logical errors is straightforward as the compiler does not explicitly indicate an error; instead, it produces a sound different from the expected one. Music inherently includes repetitive structures, which correspond to loops in programming (for statements) and conditional branches (if statements). Additionally, the structure that repeats every measure can be defined as a function in programming, demonstrating a natural alignment between music and programming.

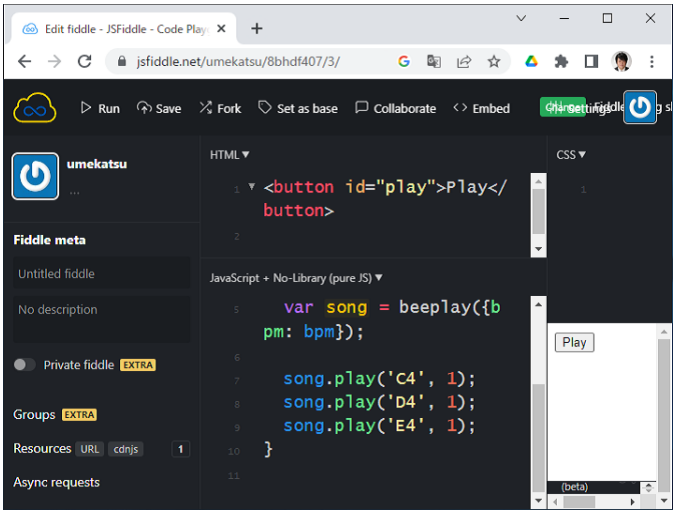

Fig. 1. JSFiddle screen

IV. Evaluation

A. Evaluation method

We conducted 12 sessions of 90 min each over a four-month period for third-year high school students aiming for admission to the Department of Informatics. These sessions focused on coding from music scores using basic programming concepts such as loops, conditional branching, and functions, employing our proposed intermediate content. We refer to students who took these classes as "experienced" and those who did not as "inexperienced." Our analysis concentrates on "Fundamentals of Programming," a mandatory first-year college course designed to teach the basics of Java and C languages, including variables, arrays, conditional branching, and loops. We examined whether there is a disparity in the mean grades between these two groups in this course. Given that the entrance examination methods for these groups differ, indicating a variance in overall academic understanding, we adjusted the programming course scores by the overall grade point average (GPA) of specialized subjects.

B. Correction by GPA

In Japan, GPA calculation methods are not standardized, and evaluation criteria vary by university. At our institution, GPA values are 1, 2, 3, and 4 for grades C (60 points or higher), B (70 points or higher), A (80 points or higher), and S (90 points or higher), respectively. If the average GPA of all first-year students is x and that of students with intermediate content experience is y, then the "Fundamentals of Programming" score p of inexperienced students is calculated using formula (1) and adjusted to p'.

$$p' = p \times \frac{10y + 55}{10x + 55} \tag{1}$$

C. Test results

Due to the non-normal distribution of the scores, a t-test was not applicable. Instead, we conducted a Wilcoxon rank sum test, a nonparametric method, to compare the average scores in "Fundamentals of Programming." The results, displayed in Table I, indicate that the experienced students, who used our intermediate content during high school, significantly outperformed the inexperienced students.

TABLE I
Wilcoxon test results

	inexperienced students	experienced students
number of students	265	32
average value	87.686	90.156
p-value	0.00585** (<0.01)	

V. Conclusion

This study demonstrated that students who engaged with intermediate content in their third year of high school achieved higher grades in programming classes during their first year of college. Future studies will analyze data from students enrolled in the academic year 2024 and beyond.

Acknowledgments

Part of the work reported here was carried out as a part of the research project "Research on e-learning for next-generation" of the Waseda Research Institute for Science and Engineering, Waseda University. Part of this work was supported by JSPS KAKENHI Grant Numbers JP24K06348, JP22H01055, JP21K18535, and JP20K03082. Research leading to this paper was partially supported by a grant from the research working group "ICT and Education" of JASMIN.

References

[1] W. Robinson, "From scratch to patch: Easing the blockstext transition," *In Proceedings of the 11th Workshop in Primary and Secondary Computing Education (ACM)*, pp. 96–99, 2016. [Online]. Available: https://doi.org/10.1145/2978249.2978265

[2] T. Tóth and G. Lovászová, "Mediation of knowledge transfer in the transition from visual to textual programming," *Informatics in Education*, vol. 20, pp. 489–511, 2021. [Online]. Available: https://doi.org/10.15388/infedu.2021.20

[3] K. Umezawa, K. Ishida, M. Nakazawa, and S. Hirasawa, "Proposal and evaluation of intermediate content for the transition from visual to text-based programming languages," *Proceedings of the 56th Hawaii International Conference on System Sciences (HICSS 2023)*, pp. 83–92, 2023.

[4] K. Umezawa, M. Nakazawa, and S. Hirasawa, "A proposal for intermediate content: Transition from visual to text-based languages," *Proceeding of the World Conference on Computers in Education (WCCE2022)*, p. 1, 2022.

[5] K. Umezawa, M. Nakazawa, and S. Hirasawa, "Comparison of biometric information during learning of visual- and text-based programming languages," *Proceedings of the 8th International STEM Education Conference (iSTEM-Ed 2023)*, pp. 1–5, 2023.

[6] K. Umezawa, T. Koshikawa, M. Nakazawa, and S. Hirasawa, "Differential analysis of heart rate, facial expressions and brain wave during learning of visual- and text-based languages," *Proceedings of the VIII IEEE World Engineering Education Conference (EDUNINE2024)*, 2024.

[7] "Jsfiddle," https://jsfiddle.net/, accessed: 22 April 2024.