Information Engineering Express
International Institute of Applied Informatics
2022, Vol. 8, No. 1, IEE527

Improving User Experience of C Programming Language Learning System for Beginners in Error Correction Learning

Shimpei Matsumoto [*], Shuichi Yamagishi [*], Kosuke Kaida [†]

Abstract

This paper aims to improve user experience of C language development environment for beginners, Hello C, which our previous researches have developed. To examine effective user experience, we conducted two kinds of experiments. The one was the experiment with the learning task of fixing syntax errors to evaluate the method to convey error messages. This experimental result suggested that the indication of the error message like scripting language may be effective even C language learning. Next, we also conducted the other experiment with the learning task of fixing I/O errors and examined the effectiveness to convey the closeness to the correct answer. This experimental result suggested that utilizing the concept of small steps could support acquiring the skill of programming.

Keywords: User experience, C language, development environment, beginners

1 Introduction

Programming learning at higher education institutions includes coding, reading comprehension, algorithm learning, and learning to improve error correction skills. Even if the lesson focuses only on coding activities, such practices aim to provide various skills, such as learning to understand grammar and algorithms. Programming requires various abilities, but learning to achieve various programming skills simultaneously is not easy for some students. In fact, some students believe that they have no programming skills even they lack only a few abilities, such as the ability to design algorithms. So, practicing education with different learning support methods according to various learning purposes is necessary to support these students appropriately. Then, various programming learning support systems are specialized for specific learning to support such practices. These systems can set learning objectives according to Instructors's intention and specialize in presenting learning tasks to achieve the goals. These are effective structures for systematically carrying out the necessary considerations because the learning activities are limited to each objective, making it easier to evaluate the results. The combination of each structure can be useful in developing comprehensive programming skills. On the other hand, learning on different platforms can be a burden for beginners. In other words, this may cause beginners to stumble due to environment settings because it forces students different operations depending

[*] Hiroshima Institute of Technology, Hiroshima, Japan
[†] Hello C Development Community, Hiroshima, Japan

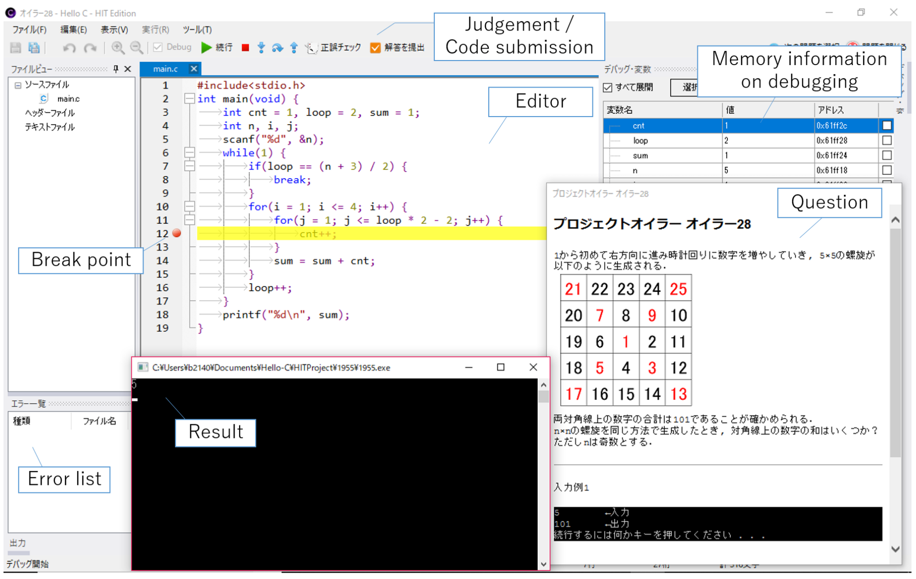
on systems. Therefore, unification of UI/UX is an important task to avoid confusion caused by such problems.

From these backgrounds, we have been developing a C language development environment for beginners, Hello C[1], including client and server systems. This paper defines beginners as "students who have less than three years of programming experience and no experience developing practical applications." Hello C is designed and developed not only for C language beginners but also for their professors. Hello C is available for a wide variety of learning support according to learning goals and aims to collect learning data according to each purpose. As shown in the literatures[2]$^{-}$[6], there are many support systems specialized for individual learning goals. As mentioned earlier, it is inappropriate for a beginner to have UI/UX independent of each piece of software. Therefore, unified infrastructure with the same UI/UX that allows beginners to receive various learning tasks with different purposes is necessary. However, due to the authors' investigations, we cannot find a C language learning support system that beginners can get various kinds of learning, and Instructorss can implement their education with a single learning base. Therefore, we have developed Hello C, which can be used as such a foundation. Hello C consists of client and server systems. Hello C client is an application available in Windows operating system environment, allowing programming beginners to touch the learning easily. Hello C server is a system for Instructorss, and it provides functions to deliver assignments, practice learning analytics, and manage data.

The purpose of this paper is to improve UX of Hello C client for beginners. For this purpose, we conducted two kinds of experiments. The first is to verify whether the method to convey error messages like scripting language is also effective in C language. Sometimes scripting languages are considered to be more appropriate for beginners. So, we hypothesized that a major factor that makes scripting languages suitable for beginners is the format of error messages. And, we thought it might be possible to improve the UX for beginners if the format of displaying error messages like scripting languages could be incorporated into the C language. We experimented with the learning task of fixing syntax errors to evaluate the method to convey error messages. This experimental result suggested that the format of displaying error messages like scripting language may be effective even C language learning. In addition to the format of displaying error messages, the feedback function is also an important UX for novice users. Then, we verified whether the feedback function to convey the closeness to the correct answer is useful or not. The closeness to the correct answer was predefined based on the output, and this information was presented during learning. In the second experiment, we gave the learning task of fixing I/O errors and examined the effectiveness. This experimental result suggested that utilizing the concept of small steps could help to acquire programming skills. Based on the above, we made the usefulness of Hello C more certain.

2 Hello C

Hello C is a programming environment specialized for the learning of the basis of C language. Hello C is an application specialized for beginners to learn introductive programming. As such, it has only the minimum functionality required for learning. Hello C does not support full-fledged application development. Hello C aims for ease of use, not high performance. Generally, when learning programming at higher educational institutions, students often use an editor at quite specialized environments such as Linux or a richer



```c
#include<stdio.h>
int main(void) {
    int cnt = 1, loop = 2, sum = 1;
    int n, i, j;
    scanf("%d", &n);
    while(1) {
        if(loop == (n + 3) / 2) {
            break;
        }
        for(i = 1; i <= 4; i++) {
            for(j = 1; j <= loop * 2 - 2; j++) {
                cnt++;
            }
            sum = sum + cnt;
        }
        loop++;
    }
    printf("%d\n", sum);
}
```

C:¥Users¥b2140¥Documents¥Hello-C¥HITProject¥1955¥1955.exe

```
5
```

変数名	値	アドレス
cnt	1	0x61ff2c
loop	2	0x61ff28
sum	1	0x61ff24
n	5	0x61ff18

プロジェクトオイラー オイラー28

プロジェクトオイラー オイラー28

1から初めて右方向に進み時計回りに数字を増やしていき，5×5の螺旋が以下のように生成される．

21	22	23	24	25
20	7	8	9	10
19	6	1	2	11
18	5	4	3	12
17	16	15	14	13

両対角線上の数字の合計は101であることが確かめられる．
n×nの螺旋を同じ方法で生成したとき，対角線上の数字の和はいくつか？
ただしnは奇数とする．

入力例1

```
5         ←入力
101       ←出力
続行するには何かキーを押してください . . .
```

Figure 1: Appearance of Hello C Client.

integrated development environment such as Microsoft Visual Studio. These kinds of software are very powerful and excellent. However, they are probably not suitable for beginning learners who are not used to operate computers. We cannot ignore this issue because such students always exist every year. We can show students' characteristics with poor computer operation skills; for example, they are not familiar with typing, do not understand file concepts, and do not know how to use an editor. For these students, a development environment with multiple functions and high performance is often a hindrance to smooth learning. Therefore, learning software with only the minimum functionality is essential to support their introductive learning. In addition, in the current programming learning environment, some troubles often occur to set up a development environment. Until now, we have seen many cases where students give up at nonessential points outside programming learning due to these kinds of troubles. These would be a critical issue for programming learning, and as a result, these can be a factor leading to the retention of learning motivation and the lack of knowledge. With this background, we started to develop Hello C. Fig.1 shows the appearance of the Hello C client. Hello C consists of two systems, Hello C client and Hello C server. These are server-client systems that work in cooperation with each other.

Instructors can create assignments using the Hello C server. In addition to the assignment title and its content, instructors can set some test cases and also various data. Students can download assignment data, including test cases, from the menu of Hello C Client without being aware of the existence of Hello C Server. After downloading, students can solve each assignment. Each assignment shows the requirements dictated by the exercises. Hello C client sends the learner's source code for each compilation and execution result (error output) to Hello C Server every time. This data is used to analyze student's comprehension or improve classes. Hello C can not only deliver practice assignments but also perform automatic scoring of practice assignments. Hello C client tests the program using test cases and realizes the automatic scoring. After the automatic scoring, the Hello C client sends log data, including the judgment of correctness, to the Hello C server. Fig2 shows the flow

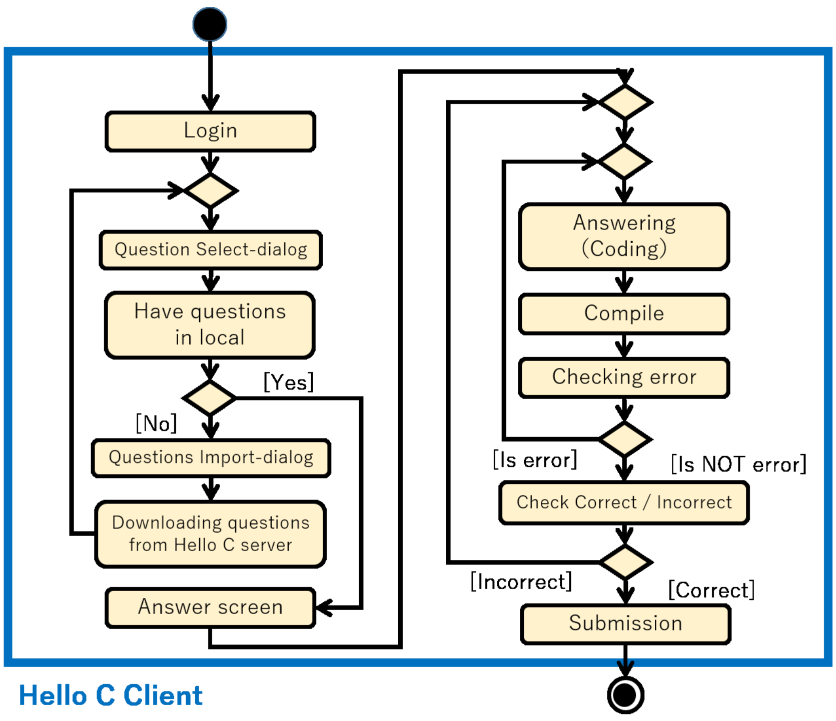


Figure 2: Learning flow

of learning using Hello C.

Hello C client uses "gcc for Windows" to compile source codes. Hello C client is developed by Visual Basic .NET Framework 4.5. Hello C server works in LAMP environment, and the target OS is Ubuntu 16.04. Its web server software is Apache 2.4.29, and its database management system is MySQL 10.1.28. Hello C server is developed by PHP 7.1.1, and its application framework is CakePHP 3.6.

3 Experiment

We conducted two experiments to provide an effective UX in Hello C client for beginners. First, we experimented with verifying whether the format of displaying compilation errors like script type language, which is often said to be easy to learn for beginners, is also effective for learning C language. Second, we also experimented with verifying whether the scaffolding based on SIEM theory, which conveys for students the closeness to the correct answer in stages according to the output result, is effective or not. In the former experiment, we gave students the learning task of correcting syntax errors. In the latter experiment, we gave the learning task of correcting I/O errors. A learning experiment and pre- post-test were conducted on the assumption that "the learning of programming was conducted by the coding-based learning which is most common lesson style, and the error occurred at first execution after completing the entry of entire source". The subjects were 16 third- and fourth-grade university students who had experience making introductive programs in C language.

Table 1: The items of the questionnaire in the experiment to verify how to display error messages that are effective for beginners

No	Question
1	Are you good at correcting all kinds of errors?
2	Are you good at finding the cause of syntax errors and fixing them?
3	Do you usually check the error statement when fixing syntax errors?
4	Was the learning task of experiment learning difficult?
5	Do you think finding and fixing the cause of all kinds of errors has a high load?
6	Do you think finding and fixing the cause of syntax errors has a high load?
7	Do you think the learning load will be smaller by limiting the number of errors displayed at one compilation to only one?
8	Do you think the learning method separating I/O errors and syntax errors will help you better understand error fixing?
9	Do you think the speed to fix syntax errors will be faster by limiting the number of errors displayed at one compilation to only one?
10	Do you think the degree of understanding to fix syntax errors will be deeper by limiting the number of errors displayed at one compilation to only one?
11	Do you think the method of displaying syntax errors only one at one compilation is more effective for beginners who are not good at programming?

3.1 Experiment on the learning task of fixing syntax errors

3.1.1 Experimental condition

We investigated the difference in learning effectiveness between the method of displaying syntax errors one by one in the same way as the scripting languages and the usual method of displaying syntax errors in C language. We gave source codes including two or more syntax errors as the learning task, and the subjects continued to learn until they corrected all syntax errors. In general, scripting languages are easy to learn for beginners. As we think there are various reasons for that, one of the major reasons might be the method of displaying syntax errors. The number of syntactical errors displayed at one time may affect the learner's cognition. Based on this assumption, the method of displaying syntax errors like scripting languages may be effective even in the learning of C language. Especially if this method can shorten the learning time and decrease cognitive load, we can say that the method is effective as a UX that makes it easier for beginners to learn C language. Therefore, this paper tested this hypothesis by focusing on two types of assignments; the one includes conditional branch and loops, and the other includes functions.

First, we divided subjects into the experimental and the control groups based on the pre-test score. We divided the subjects into two groups so that the mean scores and variances of the pre-test would be approximately equal. After determining the groups, we gave the subjects two kinds of learning tasks: assignments to learn how to write correct statements for conditional branch and loops, and assignments for functions. In the learning task, we gave the same assignments to the experimental and control groups. Also, the assignments of the learning task were the same as the pre-test. The experimental group solved them using the Hello C client, which displayed error messages only one per compilation. The control group was solved them using the usual Hello C client. Finally, the subjects addressed the post-test and answered a questionnaire on the 6-grade Likert scale. Table 2 shows the

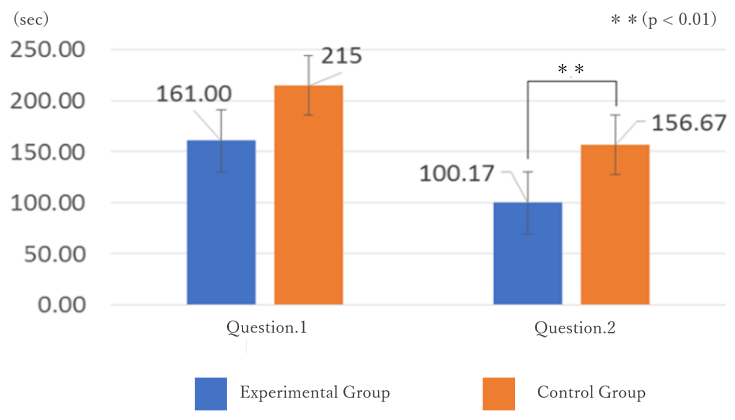


Figure 3: Answer time of experiment on the learning task of fixing syntax errors

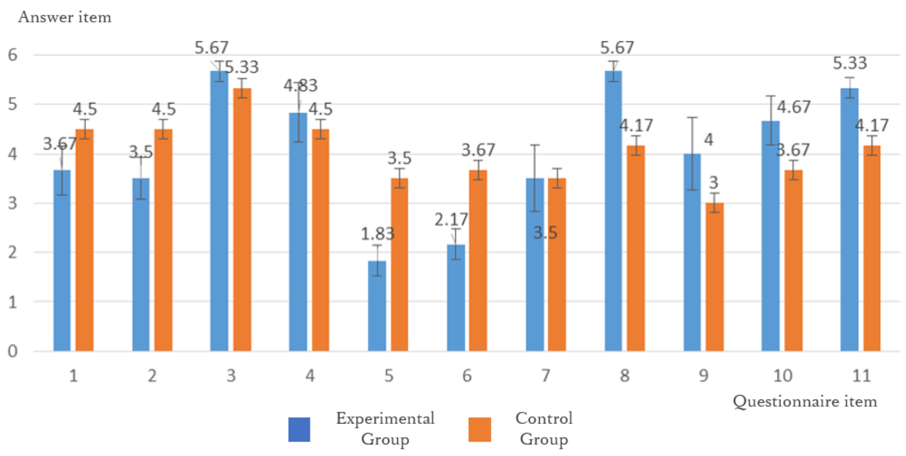


Figure 4: Questionnaire result of experiment on the learning task of fixing syntax errors

items of the questionnaire. The answers were higher for all items when “I think“ and were lower when “I don’t think“. For questions 7, 9, 10, and 11, the subjects in the experimental group answered the questionnaire based on what they had actually experienced in the experiment. On the other hand, the subjects in the control group did not experience the method of displaying syntax errors only one at a time compilation. Therefore, they answered the questionnaire based on their own subjectivity after learning the general structure of the C language.

3.1.2 Result and discussion

We measured the correct answer rate and answer time of experiment learning and analyzed these data with the questionnaire results. From these results, we evaluated the effectiveness of the proposed error display method. In the experiment learning, all subjects resulted in all assignments being correct. There was no difference between the experimental group

Table 2: The items of the questionnaire in the experiment whether the scaffolding with SIEM theory is effective or not for beginners when giving the learning task of fixing I/O error

No	Question
1	Are you good at correcting all kinds of errors?
2	Are you good at finding the cause of I/O errors and fixing these?
3	Was the learning task of this experiment difficult for you?
4	Was the question of post-test difficult for you?
5	Do you think finding and fixing the cause of all kinds of errors has a high load?
6	Do you think finding and fixing the cause of I/O errors has a high load?
7	Do you think the learning load will be smaller by conveying the closeness to the correct answer, such as "Your answer is level x when the result of specific I/O is obtained"?
8	Do you think learning separately about I/O errors and syntax errors will help you better understand error fixing?
9	Do you think the speed to fix I/O errors will be faster by conveying the closeness to the correct answer?
10	Do you think the degree of understanding to fix I/O errors will be deeper by conveying the closeness to the correct answer?
11	Do you think conveying the closeness to the correct answeris more effective for beginners who are not good at programming?

and the control group in the correct answer rate. Therefore, we analyzed the differences between the two groups based on the response speed and the questionnaire results. The results are shown in Fig.3 and Fig.4. As shown in Fig.3, the average answer time of the experimental group was shorter than the control group for both assignments. In Question 2, the experimental group was significantly shorter (Welch's t-test, $p = 0.009$, one-side) than the control group. The questionnaire results are as shown in Fig.4. As the response results were interval scale, the two groups' differences should be analyzed strictly by a non-parametric method. Here, it is enough to have a rough idea of the differences between the two groups. So, we used the mean values to get an overall trend. For questions 1 and 2, the values of the experimental group were lower than those of the control group. It suggests that limiting the number of error messages to one may reduce the confidence in error correction. Questions 5 and 6 suggest that the proposed method was effective in reducing the burden of error correction. The results of questions 8 to 11 suggest that the proposed method was an appropriate learning method.

In this experiment learning, the correct answer rate was the same in both the experimental and control groups. In addition, the correct answer time was shorter in the experimental group than in the control group. From the above, also in the learning of C language, it was suggested that displaying syntax errors one by one like scripting languages could shorten the learning time without reducing the learning effect. This result can also be seen from the result that items 10 and 11 were high in the questionnaire. From the above, it was suggested that displaying syntax errors one by one may provide a UI/UX that makes it easy for beginners to learn C language.

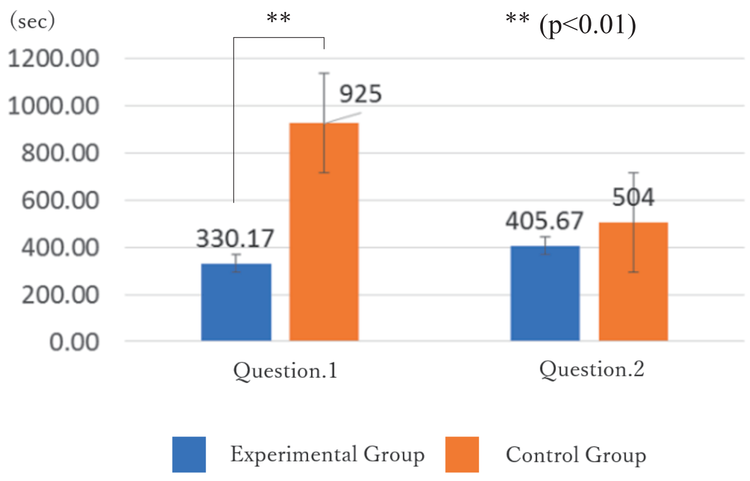


Figure 5: Answer time of experiment on the learning task of fixing I/O errors

3.2 Experiment on the learning task of fixing I/O errors

3.2.1 Experimental condition

We examined the effectiveness of small steps based on SIEM theory by giving the learning task of fixing I/O error. I/O error is a condition in which the program operates normally, but the output differs from the input due to a difference in the description or structure of the program. A learning task for fixing I/O errors means that the source code containing I/O errors is given to the learner, and the learner aims to fix it to a state close to the correct I/O. The learning with small steps based on SIEM theory[9] is a learning method that takes care to minimize failures by making the small steps to the goal. In this paper, we focus on two types of assignments: the assignments to learn conditional branches and loops and the assignments to learn functions, which are similar to the experiment of fixing syntax errors. The flow of the experiment is almost the same as the experiment of fixing syntax errors except for the content of the experiment learning. At the end of the experiment, the post-test was conducted to confirm how much the fixing ability of I/O error was improved. The flow of the learning experiment is as follows. The subjects were instructed to compile each time one part of the code is fixed. Then, under that instruction, we gave feedback only for the experimental group, such as "If this kind of output is obtained, your answer is level x (the larger the value of x, the closer the correct answer is)" along with the compilation result. We realized small steps with Hello C in this way. The subjects of the control group learned under similar instructions but without feedback. In the post-test, we presented two assignments that are the same as the learning task. After the post-test, the subjects were given the questionnaire. Table 2 shows the items of the questionnaire. The answer items were almost the same as the experiment by the syntax error correction task.

3.2.2 Result and discussion

We measured the percentage of correct answers and the answer time of the post-test. Also, based on these data but also the post-test and the questionnaire, we evaluated the effectiveness of the learning method using small steps as UX for the Hello C client. The results are shown in Fig.5 and Fig.6. The error bars in the graph represent the standard error. As for

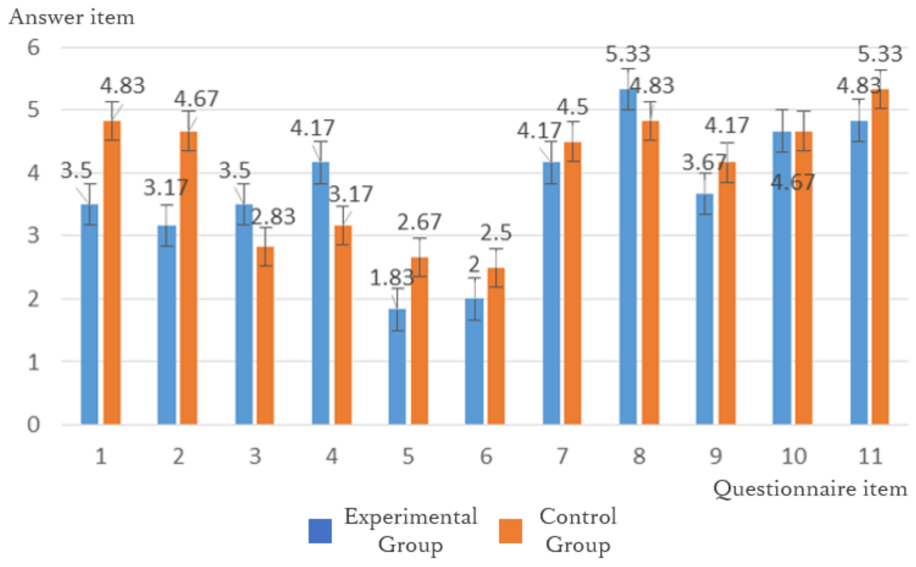


Figure 6: Questionnaire result of experiment on the learning task of fixing I/O errors

the correct answer rate of the post-test, all subjects in both groups were correct in Question 2, but the result of the experimental group was higher in Question 1. As shown in Fig.5, the average answer time of the experimental group was faster for both Question 1 and Question 2. In the post-test, the answer time was also shorter in the experimental group. In Question 1, the experimental group was significantly shorter (Welch's t-test, $p = 0.007$, one-side) than the control group. From the above, in the learning of C language, it was suggested that giving small steps may be possible to shorten the learning time without reducing the learning effectiveness. There were some descriptions in the free description column of the questionnaire: "I think that it was just fine to confirm the small steps when I was not able to advance in the middle of the answer.", "I felt it was a burden to check while looking for how much I advanced myself. Hello C will officially implement this feature." "I think giving the information on the small steps was good because the learner can always check his/her progress." These results suggest that adding a capability to give feedback based on the small step to Hello C will be effective as a UI/UX for beginners.

The questionnaire result is as shown in Fig.6. For the same reason as in the previous experiment, we used the mean values to get an overall trend. For questions 1 and 2, the values of the experimental group were lower than those of the control group. It suggests that the proposed method may reduce the confidence in I/O error correction. This result is natural because the subjects work with the hints given as small steps. The trend in questions 3 and 4 can be for the same reason. The results of the questionnaire did not indicate the usefulness of the proposed method after question 5.

4 Conclusions

This paper examined the efficient UI/UX for beginners to Hello C client. Specifically, assuming the learning to write only the source code required by Instructors, we designed the method of experiment on the learning tasks of fixing syntax errors and fixing I/O errors. In the experiment on the learning task of fixing syntax errors, it was suggested that a compilation error indication method like scripting languages may be effective also in the learning of C language for beginners. In the experiment on the learning task of fixing I/O errors, it was suggested that learning using small steps could help to understand the cause of the errors without reducing the learning effectiveness.

Acknowledgement

This work was partly supported by Foundation for the Fusion of Science and Technology 2018 and Japan Society for the Promotion of Science, KAKENHI Grant-in-Aid for Scientific Research(C), No.20K03194, and No.19K02987.

References

[1] K. Kaida, M. Ohshita and S. Matsumoto, "Developing an Editor of C language for College Students", Proceedings of Japanese Society for Information and Systems in Education, 2018, in Japanese.

[2] H. Taguchi, H. Itoga, K. Mouri, T. Yamamoto, and H. Shimakawa, "Programming Training of Students According to Individual Understanding and Attitude", IPSJ Journal, Vol. 48, No. 2, pp. 958-968 (2007) (in Japanese)

[3] S. Horiguchi, H. Igaki, A. Inoue, M. Yamada, T. Hoshi, and K. Okada, "Progress Management Metrics for Programming Education of HTML-based Learning Material", IPSJ Journal, Vol. 53, No. 1, pp. 61-71 (2012) (in Japanese)

[4] S. Cho, M. Kai, A. Kawai, T. Hino, S. Maeshima, and K. Kakehi, "Nigari - A Programming Language and Environment for the First Stage, Leading to Java World", IPSJ Journal, Vol. 45, No. SIG 9(PRO 22), pp. 25-46 (2004) (in Japanese)

[5] M. Hishina, K. Tokuoka, and K. Kawamura, "Algorithm Learning Support System with Structured Chart", IPSJ Journal, Vol. 45, No. 10, pp. 2454-2467 (2004) (in Japanese)

[6] Y. Matsuzawa, H. Yasui, M. Sugiura, and S. Sakai, "Seamless Language Migration in Introductory Programming Education through Mutual Language Translation between Visual and Java", IPSJ Journal, Vol. 55, No. 1, pp. 57-71 (2007) (in Japanese)

[7] M. Ohshita, K. Kaida and S. Matsumoto, "Constructing Development Environment of C language for Novice Learners and Implementing a Function of Basic Analysis", Japanese Society for Information and Systems in Education, P1-03, pp.5-6, 2018, in Japanese.

[8] Society for Kearning Analytics Research, https://solaresearch.org/

[9] S. Dohi, O. Miyakawa and N. Konno, "Devices of class for Introduction to Computer Programming Education in the School of Engineering Evening Division at the Department of Electrical and Electronic Engineering", Journal of JSEE, Vol.62, No.3, in Japanese.