Chen, W. et al. (Eds.) (2017). Proceedings of the 25th International Conference on Computers in Education. New Zealand: Asia-Pacific Society for Computers in Education

Encouraging System for Teaching Assistants to Advise Students during Programming Exercises

Yuuki YOKOYAMA*& Hironori EGI
The University of Electro-Communications, Tokyo Japan
*y1730120@edu.cc.uec.ac.jp

Abstract: Computer programming courses form a basic field of study for both students who major in Computer Sciences and those studying a wide range of other sciences. However, such courses present difficulties for certain students. To overcome these issues, many universities have introduced teaching assistant (TA) programs to support students' learning. However, TAs may not be able to effectively identify students having difficulties in class. Consequently, we designed a support system that enables TAs to better understand students' learning performance during programming classes. The system employs head-mounted displays, which provide TAs with a direct visual representation of students' current and historical class performances, enabling them to determine which students (if any) are facing difficulty and provide them with extra support.

Keywords: Teaching Assistants, Programming Exercises, Learning Logs, Advising

1. Introduction

Programming courses form a basic field of study for students majoring in Computer Sciences as well as those studying a wide range of other sciences. However, such courses present difficulties for certain students who may require extra support. This problem is compounded by the fact that typically, only one tutor is responsible for supervising a large number of students.

To improve this situation, many universities have introduced teaching assistants (TAs), whereby postgraduates are hired to help individual students in class. However, TAs can find it difficult to identify students who are experiencing issues. Ultimately, this can lead to students giving up and eventually failing.

To overcome this, we designed a support system that enables TAs to better understand the student learning process. The system collects performance data from the course server as well as visual feeds transmitted to the head-mounted displays (HMDs) worn by TAs, to help them determine which students are having difficulties with the class material and offer them learning support.

2. Programming Exercises and TA

Many universities run computer programming courses to teach students the basics of IT. Such courses are beneficial for students as they help them gain a better understanding of software design and structure. Classes mainly consist of students solving a series of exercises in a computer laboratory, and being taught by a single tutor. However, quite often, students experience difficulties in mastering such complex skills, especially if their background falls outside of Computer Science.

TAs play an important role in supporting students in programming classes. They have two basic tasks: (i) answering student queries about syntax or errors and (ii) identifying students who are experiencing difficulty and advising them on solutions. TAs are sourced from the student cohort and their position is closer to those of the students they are helping than as a tutor.

However, in large classes, TAs can find it difficult to identify students requiring support, and therefore, improved ways of helping TAs to determine this are required.

3. Related Work

The most effective ways of teaching computer programming are important considerations within Computer Science courses. Students learn not only the grammar of programming languages but also coding techniques and problem-solving (Robins, 2003). Students are expected to work on programming exercises and practical assignments, on which tutors and TAs give feedback afterward.

Automated assessment, which provides feedback to students, is used in Massive Open Online Courses (Staubitz, 2015); however, this technique makes it difficult to monitor individual student's learning progress.

The programming processes employed by the students represent the main way in which their learning progress is measured. To this end, a system that offers real-time, automatic monitoring, assessment, and evaluation of the code written by students as well as the production process employed by them was designed (Robinson, 2016).

Survey-based research has demonstrated that, according to TAs, verbal feedback is more effective than written feedback sent through an online system or e-mail (Rodgers, 2014). Thus, we propose a system that enables TAs to advise students based on their individual performance and learning needs.

4. Design

4.1 Requirements

The requirements for this classroom-based TA learning support system are as follows. TAs may not be able to determine which students are facing difficulties with the class material as it is difficult to monitor several learners in a large computer laboratory. The programming courses include complex elements and clear understanding requires considerable intellectual resources. The difficulties in mastering each of these elements vary among students. Therefore, by using our proposed system, TAs can identify specific elements that are not understood by particular students.

4.2 System Architecture

The courses are managed by the Learning Management System (LMS). The registered students are identified by their user IDs. Study materials are prepared and uploaded on the LMS. Achievement tests, assignments, and questions and comment about each lesson are submitted by the students. Attendance and absence are automatically checked by associating the students' user IDs with the IP addresses of the PCs in the laboratory. Our proposed system collects attendance results, achievement test scores, and assignment grades from the LMS and provides it as a visual feed to the TA via the HMD.

4.3 Implementation

Student performance data are shown via visual feeds using the HMDs worn by TAs. Figure 1 shows a photograph of a TA wearing an HMD. The optical HMD (Vuzix M100) was chosen to allow unrestricted action and movement as TAs walk around the classroom supporting students. The proposed HMD system enables more convenient communication between TAs and students than tablets.

Figure 1. TA wearing an HMD

Figure 2 shows an overview and the procedures of the system. The three steps of the procedure are as follows.

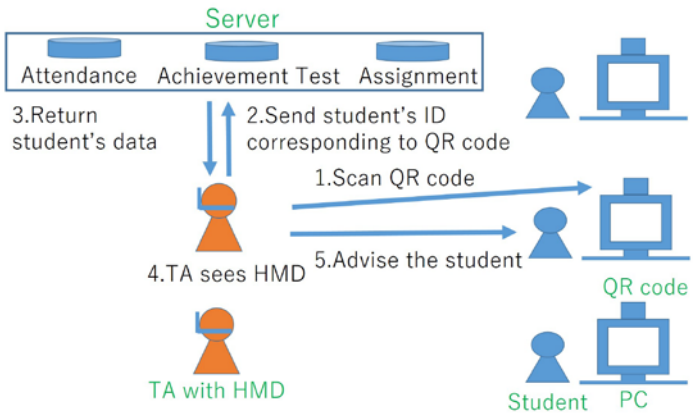

Figure 2. Overview and procedure of the system

4.3.1 Obtaining User ID of the Targeted Student

Printed QR codes associated with the IP address of each computer are provided on the back of each PC. Figure 3 shows the printed QR codes in a computer laboratory. The IP address of each computer corresponds to the user ID of the student accessing the LMS server. TAs move around the classroom during the programming class, and as the need to check a student's learning progress arises, the TA simply scans the printed QR code associated with the student. The system takes the user ID of the target student corresponding to the QR code and sends it to the server.

Figure 3. QR codes attached to PCs in a computer laboratory

4.3.2 Obtaining Three Types of Data

When the server receives the user ID of a particular student, it returns three types of data to the system for each lesson: (i) Attendance, (ii) Achievement Tests, and (iii) Assignments.

4.3.3 Visually Representing the Three Types of Data on the HMD

Figure 4 shows the visual representation of the three types of data—(i) Attendance, (ii) Achievement Tests, and (iii) Assignments—in a graph that appears on the HMD. Each block consists of the results from every lesson up to the present one with the student's learning progress represented by green, yellow, or red color. For instance, in Attendance, green signifies attendance, red is absence, and yellow is late. In the Achievement Tests, green represents high scores and red represents low scores. The colors of the boxes in Assignments are green for submitted and red for unsubmitted. Empty boxes indicate that no data are available; for instance, no tests or assignments have been given yet.

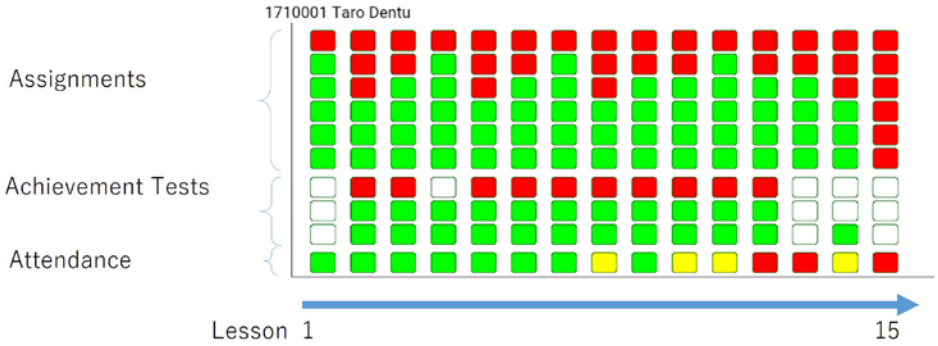

Figure 4. Visual representation of a student's class-performance data

In addition to the graphs, we designed a Seat Map that displays the class's seating arrangements together with each student's scores given in a separate graph. The Seat Map consists of two types of seating charts. The first displays the student's scores from the current lesson, while the second shows their historical scores from past lessons. Figure 5 shows an illustration of a Seat Map for students in a computer laboratory. The server calculates student's average scores in advance—which are represented by different colored boxes, where red signifies scores within the top 10% (subtracted from the average score), blue for scores within 10% of the top score (summed to the average and above), and green for scores between these two cohorts (red and blue). Seat Map can be accessed when TAs wish to monitor overall class learning progress.

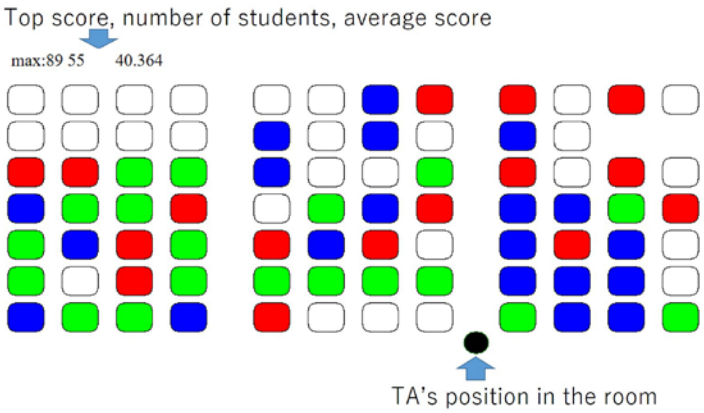

Figure 5. Seat Map showing student's performance in the computer laboratory

5. Experiment

This system was introduced to programming classes in order to evaluate its effectiveness in enabling TAs to identify students requiring support during lessons. The details of the programming classes are shown in Table 1. The system was introduced to four classes. The TAs wore HMDs and accessed student performance graphs during lessons #11 to #15 in Class A, lesson #13 in class C, and lesson #15 in Classes D and E. The number of instances where students had queries for TAs and received support were counted by observation. The lesson condition (i.e., whether HMDs were used or not) in each class is shown in Table 2. The page views (PVs) of student's data and Seat Map data were calculated from the server logs. To provide a control condition, the number of requests for TA support made by students in the other classes where the HMD system was not used (i.e., B and F–K) was also measured.

Table 1: Detail of the lecture.

Course title	Introduction to C Programming
# of teachers	1
# of TAs	2
# of students	70
Length and frequency of lessons	90 min - once a week

6. Result

The incidences of questions asked by students (# of Qs) and the consequent advice given by TAs (# of As) during the lessons are shown in Table 3. No student questions or TA advice were observed in classes F–K. The results suggest that the impact of the HMD-based system on programming classes in terms of identifying students requiring support is unclear. It was evident that TAs' behaviors varied between classes in term of offering student support to those in difficulty as certain TAs moved around the classroom and advised students freely, while others tended to sit in front of their PCs. Thus, one advantage of introducing the HMD-based system is that it encourages TAs to move around the classroom and offer support as they are required to scan the QR codes assigned to the students.

Table 2: Experimental conditions of classes.

Class	Lesson #11	Lesson #12	Lesson #13	Lesson #14	Lesson #15
A	HMD	HMD	HMD	HMD	HMD
B	Video	Video	Video	Video	Video
C	Video	N/A	HMD	N/A	N/A
D	N/A	Video	N/A	N/A	HMD
E	N/A	Video	N/A	N/A	HMD

Table 3: Number of questions and instances of advice during lessons.

	Lesson #11		Lesson #12		Lesson #13		Lesson #14		Lesson #15	
Class	# of Q	# of A	# of Q	# of A	# of Q	# of A	# of Q	# of A	# of Q	# of A
A	5	3	1	0	0	0	1	0	3	0
B	5	0	0	0	1	0	5	0	3	0
C	1	0	N/A	N/A	0	6	N/A	N/A	N/A	N/A
D	N/A	N/A	1	1	N/A	N/A	N/A	N/A	2	1
E	N/A	N/A	4	2	N/A	N/A	N/A	N/A	8	0

The number of PVs during lessons #11 to #15 in class A are shown in Table 4, and those during lesson #13 of classes A and C and lesson #15 of classes A and E are shown in Table 5. PVs from lesson #15 (class D) were not collected due to server issues.

The analysis results showed that QR codes were scanned more often by TAs in the first lesson than in the following lessons, perhaps because of the necessity of getting used to the system. In addition, the Seat Map data were viewed by TAs more often than the graphs of student performance data.

In interviews with TAs, the visual representation of student performance data used in this HMD-based system was judged to be a suitable method for identifying students in need of extra support. However, TAs did report difficulties in talking to students even after they became aware of the learner's need for support. The reasons for this require further research.

Table 4: Result of page view (PV).

Lesson	#11	#12	#13	#14	#15
PV of student's data (total)	18	1	0	0	1
PV of student's data (maximum number of a student)	3	1	0	0	1
Seat Map PV of current lesson	31	5	1	7	7
Seat Map PV of past lessons	4	7	5	10	10

Table 5: Result of page view (PV) in Lesson #13 and #15.

Class	Lesson #13		Lesson #15	
	A	C	A	E
PV of student's data (total)	0	5	1	8
PV of student's data (maximum number of a student)	0	1	1	2
Seat Map PV of current lesson	1	5	7	8
Seat Map PV of past lessons	5	13	10	10

7. Conclusion and Future Work

In summary, we designed an HMD-based system that enables TAs to gain a clear overview of students' learning progress during computer programming classes. The system collects student performance data from the course server and produces a visual representation of this for use by TAs. This method was introduced to real programming classes, allowing TAs to gain an overview of students' learning progress and performance. However, the effectiveness of this system in terms of allowing TAs to more easily identify students requiring extra support with tasks was unclear. In addition, some TAs reported that wearing the HMD throughout the lessons was uncomfortable. In light of these findings, more research is required into increasing the efficacy of HMD-based systems in identifying students in need of extra learning support. Furthermore, it would be useful to compare the effectiveness of HMD-based systems with more traditional tablet-based approaches to identifying students in need of support during a class. It appears that TAs can benefit from being trained on effective monitoring of students in order to improve their ability to detect learners requiring enhanced support within the class environment.

Acknowledgements

This work has been partly supported by the Grants-in-Aid for Scientific Research (No. 15K01025) by MEXT (Ministry of Education, Culture, Sports, Science and Technology) in Japan

References

A. Robins, J. Rountree and N. Rountree. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education (CSE), 1*3(2), 137-172.

T. Staubitz, H. Klement, J. Renz, R. Teusner and C. Meinel. (2015). Towards practical programming exercises and automated assessment in Massive Open Online Courses. *2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 23-30.

P. E. Robinson and J. Carroll. (2016). An Online System for Monitoring and Assessing the Programming Process, *Bulletin of the IEEE Technical Committee on Learning Technology, 18*(2/3), 2-5.

K.J. Rodgers, F.Marbouti, A. Shafaat, Hyunyi Jung and H.A. (2014). Diefes-Dux. Influence of teaching assistants' motivation on student learning. *2014 IEEE Frontiers in Education Conference (FIE)*, 1-8.