International Journal of Engineering Education Vol. 30, No. 3, pp. 560–565, 2014
Printed in Great Britain

0949-149X/91 $3.00+0.00

The Auxiliary Role of Information Technology in Teaching: Enhancing Programming Course Using Alice*

JACKY XI ZHANG[1],** LI LIU[2], PATRICIA ORDONEZ DE PABLOS[3] and JINGHUAI SHE[4]
[1] College of Management and Economics, Tianjin University, P.R. China.
[2] Overseas Chinese College, Capital University of Economics and Business, P.R. China.
[3] Department of Business Administration, University of Oviedo, Oviedo, Spain.
[4] Capital University of Economics and Business, P.R. China.
E-mail: jackyzhang@tju.edu.cn; Li Liu and Xi Zhang contributed equally in this paper.

Teaching and learning programming are the fun and also pain part for instructors and students in academic institutions, from the famous "Hello World" simple codes to advanced feature application software, there is a long and tough way to go for most of the students. Within limited teaching hours, instructor's presentation and explanation of the computer language are not attractive enough to raise and improve students' interest for learning proactively as students have different learning styles [1]. From the observation and the query, most students are more comfortable with learning by visual presentation—diagrams, video, animation, verbal explanation and trying things from their own interesting story. This paper describes a teaching approach for Java programming by using Alice—an innovative 3D programming environment that make it easy to help students learn fundamental programming concept in the context of creating animated movies and simple video games [2]. Evidence of students performance in learning Object-Oriented programming as a result of this approach are shown in this paper by statistical data that gathered, summarized and observed from fall semester of 2012.

Keywords: programming course; Alice; 3D programming environment

1. Introduction

Cutting teaching hours, lightening the burden of coursework from students are more emphasized nowadays and have been defined into the college education curriculum. However, the knowledge and techniques cannot be made as a short cut correspondingly. We are working on how to use the class time to stimulate the interest in learning and motivate students to practice, self-study and be involved into academic study. The ACM Computing Curricula 2001 (CC2001) report summarized four approaches to teaching introductory computer science and recognized that the "programming first" approach is the most widely used approach in North America [3]; it is the same in most colleges in China. Because of the course is for sophomores, the C language is the prerequisite of Java programming, students are familiar with simple programs in procedural language. In 2012, the faculty members started turn to use "object first" strategy to attract learning interest of programming. There are a few software tools that have strong visual and graphic component to help develop good object-oriented programming, such as Alice 3, BlueJ [4], Green Foot [5]. We selected Alice for the reason of:

- Working with an easy-to-use 3D graphics environment is attractive and highly motivating to today's generation of media-conscious students.
- The visual nature and immediate feedback of program visualizations makes it easy for students to see the impact of a statement or group of statements. Further, it makes debugging easier.
- The drag-and-drop editor prevents students from making syntax errors that are prevalent for beginners.
- The 3D modeled classes and instantiated objects in Alice provide a very concrete notion of the concept of an object and support an "object-first" approach [6].

In Overseas Chinese College at Capital University of Economics and Business (CUEB-OCC), a pilot project was conducted in the IT credential oriented class with 26 students. The objective of the investigation was to exam whether the auxiliary software, i.e., Alice, would enhance the object-oriented programming learning interest and performance of IT majors.

2. Previous research works

The seriousness of attrition in computer and information technology programs has been described in several studies. Students in Health, Engineering, Management/Commerce, Architecture and Education stayed the course better than students in Science, IT, Creative Arts, Society/Culture and Agriculture/Environment [7–12]. More than half of college students that initially declare a major in

** Corresponding author.

* Accepted 20 October 2013.

Table 1. Design a Storyboard

Component	Definition	Example
Object	A moving or non-moving character that you position and/or program to move and act	Animals, cars, people, trees
Scene	The place (or "world" in Alice) where your story occurs	Park, library, school, Home
Actions	The instructions for how each object should act in the scene	Walk 2 meters, turn left, say "Hello!"
User interaction	The ways in which the user can manipulate the objects in the animation	Keyboard commands or mouse clicks to make objects move
Design specifications	How the objects and scenery should look in the animation	Size, position, location, color

computer science or information science change their majors prior to graduation [13], or don't chose to develop their career in IT industry after graduation. Attrition occurs throughout the four years because of numbers of reasons including students' academic involvement, goal commitment, feeling of inadequacy, distress and lack of confidence. Dissatisfaction with the course content and difficult content has also been identified as important reasons for attrition. We believe the difficulties in computer science currently is related to the shift of most computer science departments from teaching introductory computer science courses using imperative languages (such as C and Pascal) to using object-oriented (OO) languages (such as C++ and Java). OO languages require that students not only learn the material for an imperative language core (e.g., assignment, decisions, functions, procedures, repetition, arrays) but also learn the additional concepts of class, object, information hiding, inheritance, and polymorphism. Many Information Science courses also include event-driven programming—yet another paradigm students must master. Clearly, the amount of material has increased but the time in class has not. Students without prior programming experience are likely overwhelmed by the breadth and depth of material, contributing to student less involvement and turn to potential attrition [6]. We keep working on finding an innovative approach to teach the fundamental concepts of object-oriented programming to our students that 73% of them don't have OO programming knowledge but just have heard of it and 19% of them even never heard about it. There are other tools besides Alice are available to assist Java learning, Alice was the one we chose as it is easy for students to design their own 3D animation. Faculty attended iLearning training provided by Oracle Academy [14] in 2012, after one month training, teachers who were experts in the areas had more confidence on making their teaching interesting and transferring knowledge efficiently.

3. Curriculum development

The objective of using Alice is to be the assistance to help students to be involved into learning programming, raise up motivation and interest, understand Classes, Methods, Control Structures, Encapsulation, Variables and other Java programming syntax. We developed tutorials as per the knowledge points, from the very beginning warming up animation program to complex projects.

3.1 Team work use storyboard

We applied a serial of tutorial for the different stages of study using Alice. Started from definition of scenarios, students in group 3 or 4 define their scenarios. A scenario is a story in the form of a problem to solve or task to perform, such as a conflict and solution in a theatrical play, a lesson to teach, a process to simulate or demonstrate or a game to play. Then use storyboard to visualize the scenario, Table 1 describes the method of designing a storyboard. Storyboards have two types, visual and textual. The storyboard can be created on paper or using digital tools such as word processing programs, paint or drawing program or presentation program. Figure 1 is an example of using visual storyboard to tell a story, then the flowchart will be

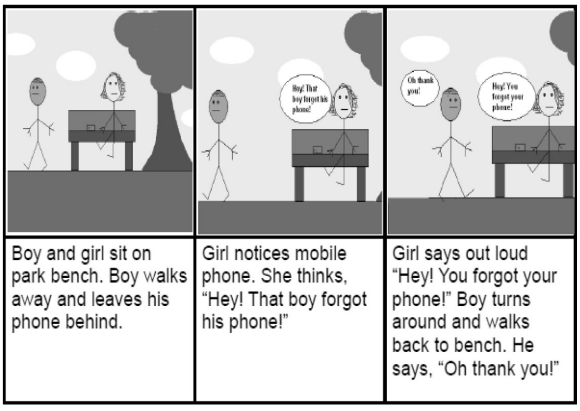

Fig. 1. Visual Story board.

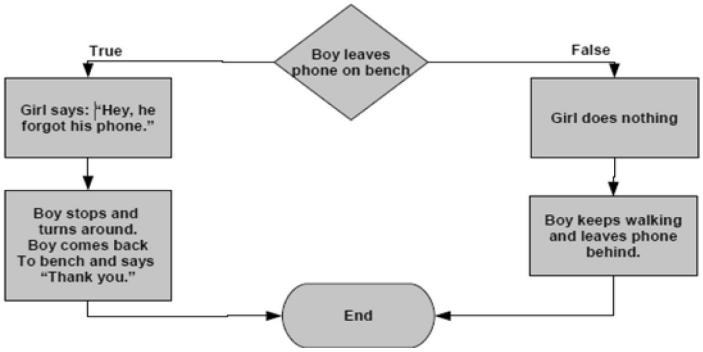

Fig. 2. Flow Chart.

Fig. 3. Myflyer, simulation of a class.

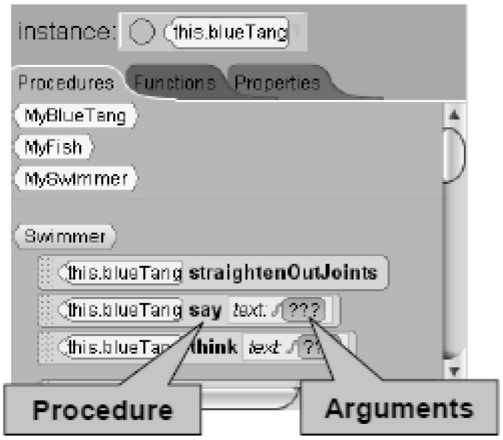

Fig. 4. Procedure and Argument in Alice.

built base on the storyboard to organize the animation's actions as Fig. 2.

Making the storyboard and flowchart trained the students to have a logical and detailed thinking-about the sequence of the events that happened in our daily life but overlooked, such as the sequence of moving include standing up, turning to right , then walking away.

3.2 Correlating java fundamentals with Alice

Based on the storyboard and flowchart, develop a scene, then the Java related knowledge such as Class, Instance and Objects are introduced. Instead of saying A class is the blueprint from which individual objects are created, the class can be visualized by choosing a folder from the gallery such as Myflyer (Fig. 3), the animal contained in the folder all have same basic properties of the flyers, two legs, two wings and feathers, and the ability to fly. Myflyer is the class, once a seagull, for example, is chosen and dragged into scene; it becomes an instance of Myflyer class. Procedures, Arguments, Control statements are the bones of Java; they are become easier by using Alice 3D animation to implement. Procedure and arguments, for example, are represented by the action definition of the objects. Base on the story students planned, the procedure are implemented by a serial of actions with proper arguments under particular control statements (Fig. 4). The integration of all the actions of an object will demonstrate an animation; bring an intuitive feeling of the program to students straight away. Similarly, control structure, conditional loop, expressions and variables are able to be presented in Alice in an animated way (Fig. 5).

4. Method

4.1 Students recruitment

Before the project introduced, we conducted a survey of 52 students, research questions included:

- Before the Java Programming class, whether they had knowledge of Java programming, 92% students had only heard of Java or no basic knowledge.
- Whether they have interest to learn and expectation. 33% have strong interest and high expectation, 54% are not very interested in learning it, but still believe it is useful and hope to have an interactive or interesting way to make the learning easier. The rest of the students have no interest at all.
- Opinions about using auxiliary software for

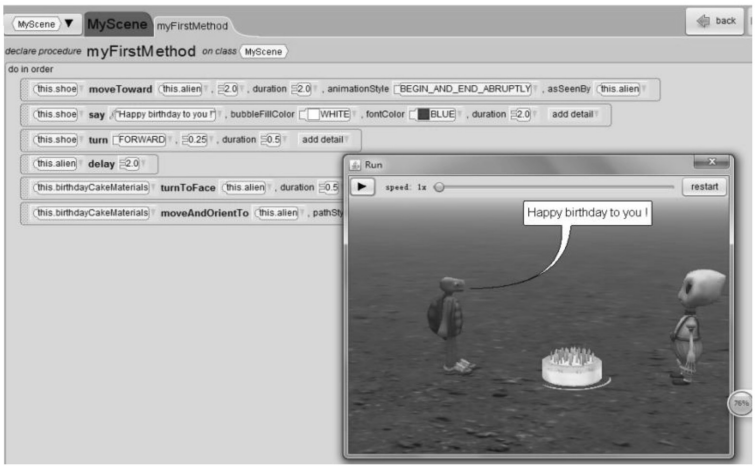

Fig. 5. Animation in Alice.

teaching computer science courses. 83% students express their interest to the approach, other students had concerns about whether the software itself is difficult to be understood, which will not be a help for the programming course.

According to the class structure and the specific learning objective, we applied Alice to one class with 26 students, defined the contents into their course curriculum. The other class with same number of students using the traditional teaching way.

4.2 Observations

Java programming was offered as 4 credits class, 2 hours instruction and 2 hours lab. Alice was applied in lab time and students were assigned group works. We have seen that students develop the strength from the following aspects:

- Team work and design sense developed. We use storyboarding and pseudo code to develop designs. Students do lots work after class and we see students writing down their thoughts, discuss and mimic the actions and behaviors, make a play. Involvement is increased.
- Materialization of classes, objects and object-oriented programming syntax. This is the most significant success of our approach. Students realized those animals, people they made in the virtual world plus the actions that simulated by Alice are corresponding to specific Java program syntax.
- A strong sense of inheritance, as students writes code to create more powerful classes.
- The concept of methods as a means of requesting an object to do something. The way to make an object perform a task is to send the object a message.
- Good intuitions concerning encapsulation. Some state information can be modified by invoking methods on an object. For example, an object's position can be changed by invoking a *move* method. But the actual spatial coordinates that represent the object's position cannot be directly accessed.
- An understanding of Boolean types. Students are prevented from dragging incorrect data-type expressions into if statements and loops [15].
- An appreciation of trial and error. Students learn to "try out" individual animation instructions as well as their user-defined methods. Each animation instruction causes a visible change in the animation. Students learn to relate individual instructions, and methods to the animated action on the screen [15]. This direct relationship can be used to support development of debugging skills.

The weakness is also found in the observation and feedback:

- The students who want to have further understanding about the code of making such animation design can't get help from Alice.
- Have no opportunities of writing code by themselves to develop the detailed sense of syntax.
- Students still need to master the usual concepts of types, variables, values, and references, as well as with the often-frustrating details of syntax.
- Most students commented that the objects in Alice gallery are not beautiful and natural, result in the animation video or game not attractive.

5. Results

5.1 Student academic performance

Table 2 illustrates the analysis of the final score from two classes in the fall semester of 2012–2013 school year. Treatment group is the group of 26 students

Table 2. Difference scores of academic performance

Analysis of Scores of Java Programming					
	Sample size	**Mean**	**St. Dev**	**T statistic**	**p-value**
Treatment Group	26	16.8961538	2.24992308	2.3939805	0.010229
Control group	26	15.4153846	2.21024537		

Table 3. Different scores of attitude

Analysis of scores of interest and motivation toward OO programs					
	Sample size	**Mean**	**St. Dev**	**T statistic**	**p-value**
Treatment Group	26	3.69230769	0.96619794	3.616283	0.000347
Control group	26	2.73076923	0.95111271		

who took part in the project; control group is the group of the other 26 students not involve Alice in learning.

We apply the two-sample t test to analyze whether the average scores from two classes are different significantly. Using a significance level of 0.05 and the p-value, 0.010229, from the analysis of the average score of the two groups, we are pretty sure that the difference between the average scores of the two groups is significant.

5.2 Attitudes

Similarly, we apply the two-sample t test to analyze whether students' interests and motivation of learning Java have been affected in two classes are significantly different. We designed the survey to 52 students in 2 class, score the motivation and interest that affected by using Alice using weight score 1 to 5. Using a significance level of 0.05 and the p-value, 0.000347, from the analysis of the average score of interests of the two groups, the difference is significant, which means using Alice improved students' interests of learning Java is significantly (Table 3).

6. Conclusion

The data presented in this paper support the effectiveness of the Alice course for improving students' performance in OO programming study and attitude toward Object-Oriented programming. The authors strongly believe that, as long as object-oriented languages are the popular language of choice in Information Management major, object-first approach is the effective way to help students master the complexities of object-oriented programming. The difference scores from two groups for content assessment showed significant difference results. The data presented here is for only one semester, sample size is also small ,the research is still on going to gather more and detailed evidence in order to provide constructive approach, motivate students in computer and information science major and reduce attrition. Additionally, we hope Alice will provide more well designed gallery source for teaching purpose.

Acknowledgments—This work was supported by the National Natural Science Foundation of China (No. 71201155), Science Foundation of Ministry of Education of China (Grant No. 13YJA630073), and Seed Foundation of Tianjin University; This work was also supported by Ms. Ying Sun, the Associated dean of Overseas Chinese College at Capital University of Economics and Business, also partially supported by Leilei Zhu and Jingning Li, the instructors of Information Management and System department of Overseas Chinese College, Capital University of Economics and Business.

References

1. J. Spurlin, Applications, Reliablity an validity of the Index of Learning Styles, *International Journal of Engineering Education,* 2005, pp. 103–112.
2. www.alice.org, Carnegie Mellon University, 2008–2013 [Online]. Available: http://www.alice.org [Accessed 2012].
3. W. D. R. P. Stephen Cooper, *Teaching objects-first in introductory computer science,* ACM SIGCSE Bulletin, 2003.
4. BlueJ, BlueJ [Online]. Available: http://www.bluej.org/about/what.html [Accessed 2013].
5. Greenfoot, University of Kent in Canterbury, UK, [Online]. Available: http://www.greenfoot.org/door [Accessed 2013].
6. S. C. Barb Moskal, Evaluating the Effectiveness of a New Instructional Approach, Alice.org.
7. J. S. Ian Olsen [Online]. Available: http://www.spre.com.au/download/AUIDFRetentionResultsFindings.pdf [Accessed 2013].
8. X. Zhang, P. O. de Pablos and Q. Xu, Culture effects on the knowledge sharing in multi-national virtual classes: A mixed method, *Computers in Human Behavior*, **31**, 2014, pp. 491–498.
9. X. Zhang, M. Yuan, Z. Chen, D. Vogel, C. Chu ,Antecedents of Coordination Effectiveness of Software Developer Dyads from Interacting Teams: An Empirical Investigation, *IEEE Transactions on Engineering Management*, **56**(3), 2099, pp. 494–507.
10. X. Zhang, D. Vogel and Z. Zhou, Effects of Information Technologies, Department Characteristics and Individual Roles on Improving Knowledge Sharing Visibility: A Qualitative Case Study, *Behaviour Information Technology*, **31**, 2012, pp. 1117–1131.
11. X. Zhang, P. de Pablos and Y. Zhang, The Relationship between Incentives, Explicit and Tacit Knowledge Contribution in Online Engineering Education Project, *International Journal of Engineering Education*, **28**, 2012, pp. 1341–1346.

12. X. Zhang, P. de Pablos and H. Zhu, The Impact of Second Life on Team Learning Outcomes from the Perspective of IT Capabilities. *International Journal of Engineering Education,* **28**, 2012, pp. 1388–1392.
13. L. E. Suter, Division of Research, Evaluation and Communication, Directorate for Education and Human Resources, 1996.
14. http://academy.oracle.com, Oracle [Online]. Available: http://academy.oracle.com [Accessed 2012].
15. R. P. Wanda Dann, Citeseer, 2000. [Online]. Available: http://citeseerx.ist.psu.edu/ [Accessed 2013].
16. H. N. Seymour, Talking about leaving. Boulder, *Westview Press,* 1997.

Xi Zhang is a Professor in the College of Management and Economics, Tianjin University. He holds a Ph.D. degree in Information Systems from City University of Hong Kong, and a Ph.D. degree in Management Science from University of Science and Technology of China. His research focuses on knowledge management, IT-based engineering education, and innovation management & policy in China-EU, with more than 40 publications in academic journals, such as Journal of Management Information Systems, IEEE Transactions on Engineering Management, Computers in Human Behavior, International Journal of Engineering Education, Behavior and Information Technology, Journal of Social Psychology, Journal of Applied Social Psychology, HICSS, ECIS, PACIS, etc. He severed as Associate editor in Decision Analytics, and editorial boards in Journal of Global Information Technology Management (SSCI), Int. J. Knowledge Management (EI), Int. J. Asian Business and Information Management, etc. He was also the committee member in international conference on knowledge management (ICKM). Currently, he is the senior member in international association on computer science and information technology (IACSIT), and member of AIS.

Li Liu is an Associate Professor in the Overseas Chinese College of Capital University of Economics and Business, the director of Information Management and System department. She leads the department that pilot the IT course teaching fully in English in main land of P.R China. She holds her master degree of University of New South Wales majored in Information Science. Her teaching and research interest focus on database management and development, IT project management, E-commerce and ITIL. She has provided leadership and expertise to faculty, improve her department to integrate multiple knowledge areas such as Financial Management, credential courses of Microsoft, Oracle and Cisco, endeavour to broaden students' knowledge and improve competitiveness.

Patricia Ordóñez de Pablos is a Professor in the Department of Business Administration in the Faculty of Economics and Business at the University of Oviedo, Spain. Her teaching and research interests focus on the areas of strategic management, knowledge management, intellectual capital, organizational learning, human resources management, information technologies and semantic web. She serves as Associate Editor of Behavior and Information Technology (BIT) journal. She is Editor in Chief of International Journal of Asian Business and Information Management. She is author of numerous papers published in leading academic journals such as Computers in Human Behavior, Journal of Knowledge Management, Information Management Systems, International Journal of Technology Management, Journal of Universal Computer Science, etc. She has published books in Springer and IGI-Global.