On Using 3D Animation for Teaching Computer Programming in Cairo University


*Computer Science Department
Faculty of Computers and Information, Cairo University
5 Ahmed Zweil St., Orman
Giza, Egypt

§The Egyptian E-Learning University
33 Mesaha St., Dokki
Giza, Egypt

Email: {ahmedsalim734, sarah hs2006, sara hamdi2008, shimoyoussef shaely, nourelshams2020}@yahoo.com
{s.khattab, m.elramly}@fci-cu.edu.eg

Abstract—In Egypt and in many developing countries, there is a high level of unemployment among youth with high and middle diplomas but with limited IT and computer programming skills or with limited ability to readjust to market changes and new work requirements. This is sometimes attributed to a psychological barrier (e.g., fear or intimidation) to IT and computer programming. In this paper, we study the impact of a novel teaching method, namely the Alice 3D learning environment, proposed elsewhere for teaching computer programming concepts to novice users in an easy and attractive way.

We conducted an experiment to measure the effectiveness of Alice by using it to teach programming concepts to first-level students in the Faculty of Computers and Information, Cairo University in the academic year 2008-2009. This was done before introducing them to a high-level language. We then conducted a survey to measure the level of acceptance and benefit attained by the target students. The most important result of our study is that learning with Alice was a beneficial experience to students who have first-time exposure to computer programming. However, Alice was boring to students who have previous programming background. To the best of our knowledge, this study is the first of its kind in Egypt as Alice was not taught before here, and we hope that our study helps the adoption decisions of Alice or similar tools in other schools and universities.

Keywords— teaching; computer programming; 3D animation; Alice.

I. INTRODUCTION

Many countries suffer from a shortage of skilled Information Technology (IT) workforce; Bill Gates requested in his testimony before the Committee on Science and Technology of the U.S. House of Representatives that more skilled IT workers be allowed to enter and work in the US IT market. This was mainly because every year around 100,000 new IT jobs need to be filled and local skilled workforce is not enough [1].

The main barriers to delve into the IT world is psychological; many youth see IT (and computer programming in specific) as a remote field that can be handled only by “geeks.” To address this problem, governmental and non-governmental entities have developed programs and initiatives for training in IT field through a number of grants and specialized technical institutes. Distinguished in these efforts is a growing trend to make programming and IT concepts attractive and “fun” for kids and teenagers in order to prepare them technically and psychologically from an early age to be ready to effectively enter the IT job market when they grow up.

Teaching programming to novice students using traditional methods is a difficult task [3]–[5]. Many research efforts have been directed to improve the teaching methodologies used in introductory programming courses. For instance, Alice was developed by Carnegie Mellon University (CMU) to teach programming concepts using an easy, attractive interface that enables novice programming students to focus on the concepts and bypass tedious syntax errors and other programming details that are annoying and frustrating to many students [2]. Using Alice, students can build 3D animations and games while learning the basic concepts of flow control, object-oriented abstractions, comments, and other programming concepts. Teaching programming with Alice has been offered in top-ranked universities and schools around the world [6].

In this paper, we present results of an experiment that we conducted within Cairo University in Egypt to measure the effectiveness of teaching programming concepts using Alice. We taught Alice to junior students in the Faculty of Computers and Information in the academic year 2008-2009 within an introductory programming course (Programming-1). We conducted a survey-based study to measure how much Alice, as a teaching tool, was accepted among students and how much benefit the students gained from learning to program using Alice.
Our main results are: (1) 83% of surveyed students found that Alice was instrumental for them to learn programming concepts either partially or fully and (2) Student comments indicated that Alice was fun to students who are learning programming for the first time, whereas it was boring to students who have a background of programming. Although similar studies have been conducted in other regions of the world [7], our study, to the best of our knowledge, is the first in Egypt. We believe that our results will be beneficial when Alice is applied in other schools and universities and in designing and implementing initiatives and programs to teach programming and IT skills in general to youth and kids.

The rest of the paper is organized as follows. In the next section, we present a brief background on the challenges faced by novice programming students and Alice which was designed and developed to overcome these challenges. We conclude the section by a statement of the hypothesis we tested in this study.

II. BACKGROUND AND PROBLEM STATEMENT

In this section, we present a brief background on the main challenges faced by novice programming students and Alice which was designed and developed to overcome these challenges. We conclude the section by a statement of the hypothesis we tested in this study.

A. Difficulties in Learning Programming

To a novice programming student, the world of programming may appear mysterious and confusing (similar to the wonderland that Alice Liddell found all unfamiliar) though it does not have to be like that if the principal concepts of programming were carefully and clearly explained and understood without the intricacies of the programming languages and environments.

Taking the novice all of a sudden into strange words, such as do-while, if-then-else, switch-case, and for, usually gives the student a confusing image of programming that often results in poor performance and low marks and even dropping out of the IT field all together. The abstract concepts behind these words need to be explained to students in a tangible way that they can feel, visualize, and relate to. They should be given the chance to test each of these concepts separately to grasp the meaning and effect of each one of them. However, it is not always the case that students have this chance. On one hand, the way of teaching programming is “dry” and full of syntax and syntax errors. Second, the course schedule does not allow the teacher and the students to spend enough time on the concepts before delving into details.

The outcome is that many students develop a hazy image of the basic concepts; they may use them but without understanding. More importantly, they may use programming constructs without the ability to think abstractly. All in all, many novice programming students may end up lacking the ability to express their creative thinking in terms of programming abstractions because simply they do not grasp these concepts on an abstract level. They may write code but without really understanding each and every line in their code [3].

The above-mentioned challenges of learning programming prevents many students from entering the IT industry; some of them may actually get their degrees but without mastering programming and consequently without the ability to get the jobs they are trained to have. This results in a gap between the skills needed for many available IT jobs both locally and world-wide and the skills that IT graduates have.
B. Alice

Alice is a program designed by CMU and offered for free to teach programming concepts through making 3D animations [2]. Alice makes it easy to create 3D animations and games and even to publish them on the WWW. Figure 1 shows a sample screen-shot of Alice’s user interface. Using Alice, students create “scenarios” using drag-and-drop of constructs that reflect the basic programming abstractions, such as looping, selection, comments, parallel execution, and others. By that, Alice helps novice programming students explore the world of real programming with more confidence and with a solid understanding and imagination of the programming principles, helping to overcome the challenges and difficulties faced by novice programming students.

C. Problem Statement

In this paper, we aim at checking the validity of the following hypothesis: Teaching Alice to novice programming students improves their understanding of programming concepts. To test the hypothesis, we conducted an experiment of teaching Alice to the students of an introductory programming course. The students were mainly junior students in the Faculty of Computers and Information in Cairo University. We then surveyed a sample of the student population. The details of the experiment and the survey are presented in the next section, and the survey results are presented and analyzed in Section IV.

III. METHODOLOGY

In this section, we present our methodology in the study we conducted to measure the effectiveness of Alice in teaching computer programming concepts. We taught Alice to the students of the first level in the Faculty of Computers and Information in Cairo University in the academic year 2008-2009 within the Programming-1 course, which is mainly focused on teaching programming in C++.

We also conducted a survey to evaluate the acceptability of Alice among the students and their benefit from the tool. In what follows, we describe our teaching methods and detail what the survey covered and how we conducted it.

A. Teaching Alice

Our teaching of Alice consisted of classroom lectures, hands-on labs, a project, and one-on-one tutoring to students that needed it. We taught Alice before teaching C++, whereby we offered two lectures. The first lecture introduced Alice using a video of what Alice can do and described some of the basic techniques of using Alice, such as how to save and open Alice created files (called worlds), how to go through Alice tutorials, and how to do simple tasks involving loops and conditions. In the second lecture, we described how to do more advanced techniques in Alice, such as camera movement, object placement, variables and parameters, events, etc. We also offered three supervised lab sessions and concluded the sessions with a project to allow the students to practice what they learned and allow their imagination to drive their learning.

The project required from the students to build a movie or game to express their views on a problem they face in their day-to-day life, in their country, or in the whole world. We also offered one-on-one tutoring sessions to students who had trouble working with Alice; the problems were mainly because of students choosing method or variable names with Arabic-script in Alice, getting confused with how to open a created Alice world (simply double-clicking on the created file did not automatically open the file in Alice), and version mismatch between the Alice 2.0 and Alice Storytelling versions [2].

The project was a very rewarding experience for us and for the students. Many students have shown a high level of creativity in creating movies and games that express their views on many problems, such as traffic congestion, homeless children, noise, business frauds, and others.

Some students even moved a step further and suggested solutions to some of the problems that are globally recognized, such as air and water pollution and wars both in the middle-east and in other places of the world. This particular outcome was rewarding to us because besides achieving our main goal of teaching programming concepts, Alice allowed the students to suggest practical and distinguished solutions for real problems and express the solutions in 3D animations and games.

B. Survey

After teaching Alice and collecting and grading the student projects, we conducted a survey to measure the following: (1) the extent to which students benefited from Alice to understand programming principles, such as sequential execution versus branching, objects, methods, functions, variables, and parameters and (2) the extent to which the students enjoyed Alice and the acceptability and usability of Alice itself.

We created the survey as a form on Google Docs [8] to be easier to administer and distribute to students; distribution was done by sending target students a link to the web page. Sending the link to the students provided an acceptable level of security against unsolicited others contributing to the survey and a level of privacy to allow the students to answer freely (students can answer from the privacy of their homes, and we did not record any identification information).

The number of students who participated in the survey was 55 out of 369 students who turned in the project. We compared the statistical distributions of the project grades of the students who participated in the survey (Figure 2-b) and all the students who turned in the project (Figure 2-a). As shown in Figure 2, the survey covered all grade levels and the two distributions look almost the same; for instance, the majority of students got 6 out of 8 in both distributions. About 62% had little background on programming, and 38% were exposed to programming for the first time.

IV. RESULTS

In this section, we present the results of our survey as shown in Figure 3.
Fig. 2. Student grade distribution for (a) all students in the course and (b) surveyed students.

First (Figure 3-a), 83% of surveyed students found Alice beneficial in learning programming concepts (fully or partially). A related result that is not shown in the figure is that about one third of students reported that they learned some basic programming concepts before learning Alice. This result suggests that novice programming students even with some background benefited from Alice in learning and understanding programming concepts.

Second (Figure 3-b), 58% of the surveyed students found that learning Alice itself was extremely easy, only 2% reported that learning Alice was extremely difficult, and the remaining 40% reported an experience between the two extremes.

Third (Figure 3-c), 31% of the surveyed students learned Alice completely on their own, 6% with the help of peers only, and the rest with the help of the teaching team only or with peers.

Fourth (Figure 3-d), 39% of the surveyed students found Alice as fun, 25% found it boring, and the rest were neutral.

The free-text comments, which we allowed the surveyed students to enter and of which we report samples in the appendix, provided more details about the outcomes of our experiment. The comments were more in favor of Alice from students who did not have previous programming exposure. Those students were more enthusiastic to learn Alice. The students who had previous superficial exposure to programming had slightly less enthusiasm in learning Alice.

Very few of the surveyed sample had a more in-depth exposure to programming, and those students did not show any enthusiasm to learning Alice, a behavior that we conjecture was because Alice does not provide those students with any new knowledge. Most surveyed students liked Alice, viewed it as fun, and reported that they would advise any novice programming student to learn Alice first.

V. DISCUSSION

In this section, we discuss our experience with teaching Alice. We report on the main challenges that we faced, the lessons we learned from survey results and from our direct interaction with students, and some suggestions on how to adopt Alice in future teaching settings.

A. Faced Challenges

The main challenges we faced were: (1) the limited time for students to develop their projects: students had only two weeks to work on their project after they learned Alice, and this was because Alice was taught as a short module in the course; (2) the limited resources in terms of computers; (3) the limited characters in Alice, which were remote to students’ culture, and thus limiting students’ ability to express their ideas in their projects; and (4) a few technical difficulties with Alice, such as inability to treat Arabic-script names of files and objects and the need to change screen resolution to run the program in some computers.

B. Learned Lessons

By reflecting on the previous results and discussions, we came up with the following lessons: (1) teaching programming concepts using 3D animations make it easier for most students to grasp these concepts; (2) learning Alice itself was easy to
many students either on their own or by getting help from peers and the teaching team; and (3) the students with no programming background are the ones who benefited the most from Alice and found it fun, and the more programming experience students have the less benefit they gained from Alice.

We conclude that the study hypothesis is correct: using Alice to teach programming concepts to novice programming students was effective and acceptable. Moreover, although not quantitatively measured in this study, teaching with Alice resulted in a noticeable improvement of student understanding of programming concepts.

C. Suggestions

In this subsection, we discuss a number of suggestions that we think will help future adoption of Alice in other teaching settings. First, we think that Alice teaching to college students is best placed before students enter the university (as a Summer preparation course or as part of the Introduction to Computer Science course) with a course project to maximize the benefit of Alice when students have no previous exposure to computer programming.

Second, making direct connections between the programming concepts that students practiced using Alice and the programming concepts taught as part of the main programming language in the course, such as C++ or Java.

Third, designing and implementing workshops and training programs for school-age kids to give them a positive image about programming as a pleasant, fun topic.

VI. Conclusion

In this paper we studied the application of the Alice 3D environment to teach programming concepts using 3D animations to a group of students learning programming mostly for the first time. We conducted a live experiment to measure the effectiveness of Alice and its ease of use. We believe that our results are encouraging to adopt Alice, as a teaching tool of computer programming, for students learning programming for the first time, such as kids, teenagers, and junior college students.

In the future we plan to (1) localize Alice to the Arabic world by arabizing the user interface and adding characters that relate more to students in Egypt; (2) repeat the experiment using newer versions of Alice (e.g., 2.2 and 3.0), which enable more features such as exporting created animations as videos and automatic integration with Java Integrated Development Environments [2]; (3) give more time for students to learn Alice by offering it as an optional Summer course before the Programming-1 course; and (4) repeat the experiment using other tools for teaching programming to study their impact and further strengthen and generalize the results of this study.

REFERENCES


APPENDIX

STUDENT FREE-TEXT FEEDBACK

In this appendix, we present a sample of the free-text responses we got from the surveyed students. The student input was in the Arabic language, and here we offer a translation to English of some of the students’ opinions.

“...it was a great experience and it helps understand programming concepts...”

“...it was fun and helped us love programming and enjoy the rest of the course...”

“...the experience of learning Alice was really fun because through this program, I was able to grasp all concepts of programming. Alice conveyed these concepts to me in an easy, simple way and helped me to cooperate with my peers and help them...”

“...frankly speaking it saved us a lot of psychological pressure if the first assignment in the course would be writing C++ code...”

“...it was fun to some extent...”

“...it was not a good experience. It wasted time because I did not learn anything new in C++. Alice is very slow and boring. 3D programs like this should be fun but this one was very boring. I do not know what is the relationship between Alice and programming. I do not want this experience to be repeated.”

“...it is a good experience but to those who learned Alice before C. For those who know C or Java, learning Alice is boring.”

“...in my opinions this experience was not needed at all...we could have used its time to learn new things in C such as graphics or any other thing...Thank you”