2012 4th International Conference on Education Technology and Computer (ICETC 2012)
IPCSIT vol. 43 (2012) © (2012) IACSIT Press, Singapore

Bridging the Gap: the Role of Mediated Transfer for Computer Programming

Jacqui Chetty [+], Glenda Barlow-Jones

University Of Johannesburg

Abstract. The objective of computer programmingis thatstudentslearn how to develop solutions in computer programing languages, such as Java. However, to develop such solutions students need to be able to solve problems. Therefore,problem solving is a critical skill that needs to be acquired. As problem solving and computer programming is difficult, universities worldwide make use of interactive tools, such as Scratch, Alice and Greenfoot to provide a user-friendly, visual andcomfortable computerprogramming environment. The aim of such tools is for students to develop computer programming concepts informally. However,if students are to become competent computer programmers, theymusttransfer the programming concepts learnt from such tools, to formalised computer programming languages.This paperexaminesthe extent to which mediated transfer is an effective pedagogy to transition students. The results indicate that the transition may not be as seamless as was first expected.

Keywords: mediated transfer, Scratch, problem solving, interactive tools

1. Introduction

A study commissioned by the Information Technology Association of America (ITAA) indicated that in 2001, the United States of America (USA) alone experienced a shortfall of 425000 Information Technology (IT) skilled workers. This shortfall increases exponentially each year [1]andcomputer programming is one such technical skill where a shortfall can annually been seen[2]. This shortfall is often due to the complexities associated with learning the fundamental concepts associated with computer programming[3].

It has therefore become evident that well-trained computer programmers are needed to decrease the shortfall. However, this is not an easy task, as computer programming is a process of understanding how to solve problems; learning the syntax of a programming language; and acquiring the ability to implement a solution. It requires that students are able to think critically and creatively, in order to solve problems [2, 4].

The IT industryrelies on universities and colleges worldwide to develop degrees and diplomas that can provide computer expertise. However, at the University of Johannesburg (UJ), South Africa many students may not meet the demands of this industry, as they are often under prepared for post-secondary educational challenges[5].

Consequently, UJ is continuously exploring and researching innovative pedagogies, paradigms and interactive tools to assist these students. This paper describes how an interactive tool, namely Scratch [6] can be used to develop students algorithmic problem solving skills,as algorithms form the foundation on which programs are built [7]. Although Scratch can assist students with algorithmic thought processes [6], Scratch is not a suitable programming environment to develop computer programming solutions for business applications, in the IT industry.

Therefore, if students are to be successful in this industry, they need toadvance towards solving problems using formalised problem solving tools, such as pseudo code and flowcharting, and computer programming languages, such as Java. This paper analyses the extent to which students at UJ are able to transfer learning from one context to another. The findings illustrate that the transition is more difficult than expected.

[+] Corresponding author. Tel.: +(27 11 559 1177); fax: +(27 11 559 1239).
E-mail address:(jacquic@uj.ac.za)..

2. Interactive Educational Tools

In order to increase the pass rate, lecturers often need to make use of interactive tools to bridge the gap for students, so that they are able to solve problems expected of them. To assist students,interactive tools, such as Scratch, Alice[8] and Greenfoot[9]can be adopted. These tools provide a visual computer programming experience that assists students in developing problem solving skills and essential computer programming concepts, without having to learn the complexitiesassociated witha computer programming language, such as Java. Scratch is a popular interactive tool used by universities worldwide to support and bridge the gap for students [6, 10].

2.1. Scratch

Scratch is a "media-rich programming environment" that was developed by the Massachusetts Institute of Technology (MIT)'s Media Lab. It allows students to design, develop, create and test Scratch-based projects [6]. Problems are presented to students,where students developalgorithmic solutions in the form of stories and games. Algorithmic solutions are solutions based on a well-ordered collection of unambiguous and effective statements that, when executed, produces a result [7].

In order to develop Scratch projects the Scratch interfaceallows students to develop algorithmic problem solving solutions [7] by dragging and dropping built-in instructions onto the Scripts Editor, as seen to the left of figure 1 below. This action is known as "snapping" instructions together like Lego blocks. Once the solution is complete the Scratch project executes on a stage, seen to the right of figure 1 below.

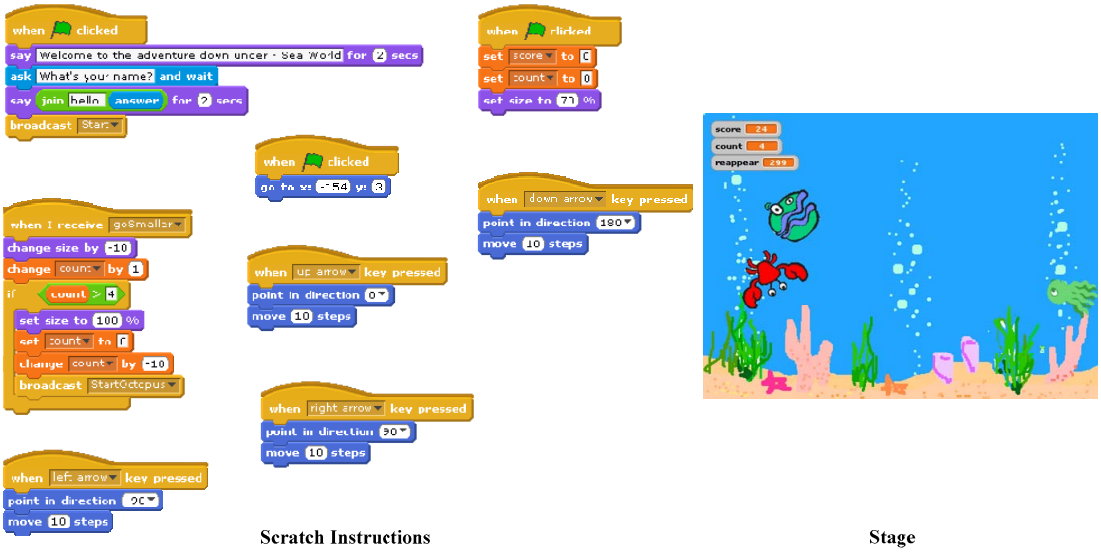

Fig. 1: Scratch Instructions

Although Scratch provides students with an opportunity to develop algorithmic skills, if students want to further enhance their algorithmic problem solving skills and become computer programmers, it is essential that students are able to solve problems using formalised problem solving techniques. Therefore, students needed to transfer their Scratch learning to another context. This can be realised through mediated transfer.

3. Mediated Transfer

Educational theorists are in agreement that learning should be transferrable from one context to another[11], also known as mediated transfer. Mediated transfer is an active research field [12] and can be defined as the ability of students to either enhance or undermine learning that took place in one context, in relation to another context [12]. It uses various techniques, such as bridging and hugging, which allow the transfer to take place [11]. Whereas bridging assists students in building a bridge from the context in which a concept was learnt into other contexts, hugging means mirroring an example in one context to the same example in another context. For example, a Scratch program can be mirrored or compared instruction by instruction, to the same example in Java.

Although research indicates that lecturers can teach for transfer [1, 11, 13] it is also quite common that students are unable to achieve transfer of learning[11]. It must therefore not be assumed that knowledge is

easily transferred[12]. To this end, the transfer of learning can be a positive experience (increases performance) or it can result in a negative (undermines performance) one.Furthermore, transfer of learning is more prone to take place where the differing contexts are closely related, also known as near transfer, as opposed to far transfer, where the contexts are dissimilar [12].

4. From Learning To Application

The department of Applied Information Systems (AIS) at UJembarked on a pilot study, where the aim of the study was to provide students with an opportunity to slowly progress towards developing computer programs. This progression took place by(1)introducing an interactive tool;and then (2)introducing pseudo code and flowcharting.The first step was to introduce the students to an interactive tool, namely Scratch.

4.1. Scratching The Surface

Scratch was deployed as the interactive tool to assist students in developing analytical thought processesand problem solving skills. The objective is to provide students with an opportunity to develop their problem solving skills by being exposed to a variety of problems through Scratch. Scratch was applied for the first six weeks of the semester. It was students' first exposure to algorithmic-based solutions and it was met with great enthusiasm by students and lecturers alike. This enthusiasm filtered through to assessments and the assessment resultaveraged60%.

Once the students could solve problems using Scratch, it was hoped that the learning could be transferred from the Scratch context to an algorithmic problem solving context. As transfer of knowledge is integral to teaching-and-learning [11] it was thought that problem solving skills applied in Scratch would assist students to develop solutions using pseudo code and flowcharting.

4.2. From Scratch To Formalised Problem Solving

The students were given an opportunity to understand how to develop solutions using pseudo code, seen to the left of figure 2 below, and flowcharting.The transfer of learning consisted of students being given similar problems to those seen in Scratch, as seen to the right of figure 2 below. The technique used for the transfer was "hugging". Once students got used to the idea of developing solutions using formalised problem solving tools, it then became mandatory for them to develop solutions using such tools.

Students struggled to develop solutions formally. Instead of the learning being transferred from one context to another, students were bewildered and were not sure how to design and implement solutions using pseudo code and flowcharting. Consequently, the first set of learner assessment results waspoor with the average totalling 29%. However, after two additional tests the average increased to 34%. Additionally, students' results for problem solving in the exam averaged at 43%. These results were however, much lower than the Scratch test results. Even though students had previously solved such problems using Scratch it was clear that students struggled to solve the same problems using more formalised problem solving techniques. The transition from interactive tool to formalised problems solving was not as seamless as was expected.

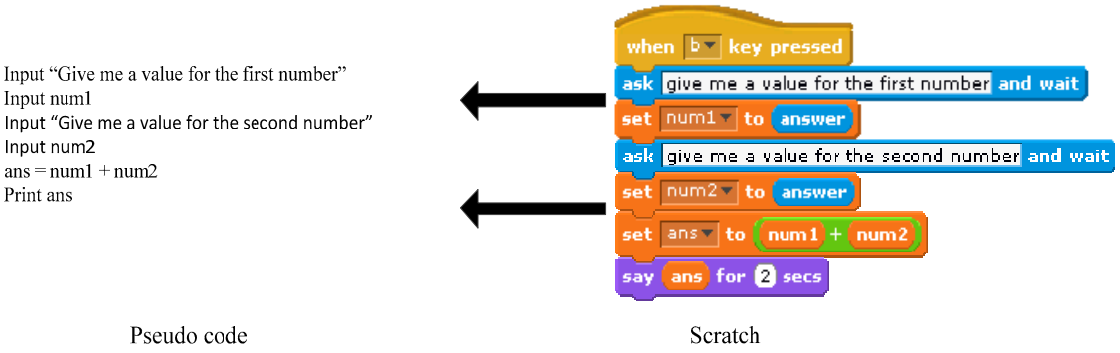

Fig.2: Mediated Transfer Using Hugging

After having much success using Scratch, especially as research indicates that interactive tools do assist students in learning fundamental computer programming concepts, the findings below indicate that computer programming concepts cannot easily be transferred from one context to another.

5. Findings

Data related to students Scratch test results and their problem solving exam results was collected and analysed using a box-and-whisker plot in combination with a paired-samples t-test[14]. The findings reveal the following:

5.1. Box-and-whisker Plot

As a box-and-whisker plot reveals how the numbers in a distribution are arranged, and where the majority of such numbers are located[14], the researchers felt it appropriate to illustrate the findings using plots. Figure 3 below illustrates a box-and-whisker plot, where the first plot indicates students' Scratch test results, and the second plot indicates students'problem solving exam results. These results indicate that the lowest score for the Scratch test was20%, the highest score was 93%, and the median score (50th percentile) was 60.35%. The lowest score for the problem solving exam was5%, the highest score was 80%, and the median score (50th percentile) was 42%. The median decrease in scores was 18.35%.

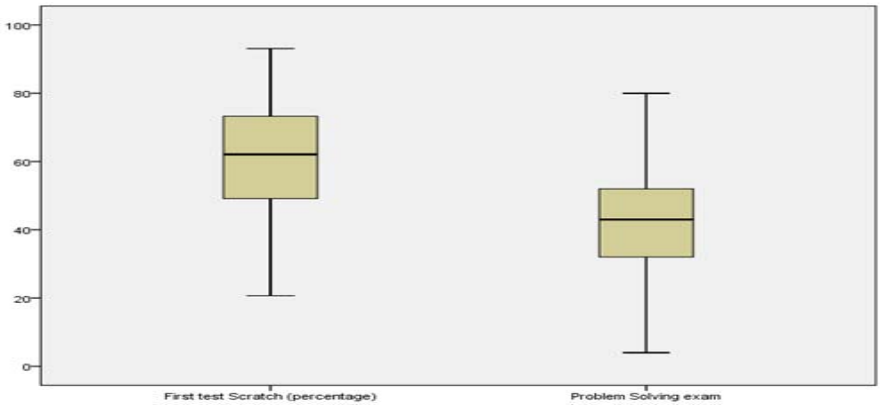

Fig. 3: Box-and-Whisker Plot

5.2. Paired-samples t-test

A paired-samples t-test, as illustrated in table 1 below, was conducted. There was a statistically significant decrease in the problem solving scores (N = 84, M = 43.17%, SD = 15.55) compared to the Scratch scores (N = 84, M = 60.42%, SD = 17.96), t(83) = 7.996, P<.000 (two-tailed). The mean decrease in scores was 17.25 with a 95% confidence interval ranging from 12.95 to 21.53. The eta squared statistic (.43) indicates a large effect size.

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	Scratch_Percentage First test Scratch (percentage)	60.4269	84	17.96349	1.95998
	Prob_solve_exam Problem Solving exam	43.1786	84	15.55492	1.69718

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	Scratch_Percentage First test Scratch (percentage) - Prob_solve_exam Problem Solving exam	17.24836	19.76983	2.15706	12.95804	21.53867	7.996	83	.000

Table 1: Paired-Samples T-Test

Therefore, as there is a large discrepancy between the median of the box-and-whisker plots and the mean of the paired samples t-test, the results illustrate that transfer of learning did not easily take place.

6. Conclusions and Recommendations

Although educational theorists indicate that learning should be transferrable, and we almost expect transfer of learning to take place when formalised teaching-and-learning pedagogy is applied, it is important to acknowledge that this may not necessarily be the case. In this instance, using a pedagogical approach such as mediated transfer and an interactive tool such as Scratch was not enough to assist students in making the transition.To this end, further research will investigate whether factors, such as hands-on practice, learner-centric pedagogies, increased collaboration amongst students, and metacognition may further improve the transition from Scratch to formalised problem solving.

7. References

[1] Mendoza, R.A., Ellis, H.J.C., *KNOWLEDGE TRANSFER IN EDUCATION: A PROJECT-BASED PEDAGOGICAL APPROACH TO BRIDGING THE APPLICABILITY GAP*. Consortium for Computing Sciences in Colleges, 2003. **18**(3): p. 13.

[2] Kak, A. (2009) *Teaching Programming*.

[3] Wulf, T. *Constructivist Approaches for Teaching Computer Programming*. in *SIGITE'05*. 2005. New Jersey, USA: ACM.

[4] Sprankle, M.H., J., *Problem Solving & Programming Concepts*. 8th ed. 2009, New Jersey: Pearson Education. 502.

[5] Jansen, J. *I would seriously consider not sending my child to school in SA*. 2012 24 February 2012]; Available from: http://www.moneyweb.co.za/mw/view/mw/en/page292681?oid=562851&sn=2009%20Detail.

[6] Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E., *The Scratch Programming Language Environment.* ACM Transactions On Computing Education, 2010. **10**(4).

[7] Schneider, G.M., Gersting, J.L., *Invitation to Computer Science*. 6th edition ed. 2013: CENGAGE Learning. 855.

[8] Dann, W., Cooper, S., Pausch, K., *Learning To Program With Alice*. 2006: Prentice Hall.

[9] Kolling, M., *Introduction To Programming With Greenfoot*. 2009: Pearson education.

[10] Fesakis, G., Kiriaki, S. *Influence of the Familiarization with "Scratch" on Future Teachers' Opinions and Attitudes About Programming and ICT in Education*. in *ITiCSE'09*. 2009. Paris, France: ACM.

[11] Dann, W., Cosgrove, D., Slater, D., Culyba, D., Cooper, S. *Mediated Transfer: Alice 3 to Java*. in *SIGCSE'12*. 2012. Raleigh, North Carolina, USA: ACM.

[12] Perkins, D.N., Salomon, G., *Transfer of Learning*, in *International Encyclopedia of Education, Second Edition*. 1992, Pergamon Press: Oxford, England.

[13] Fadel, K.J., Brown, S.A., Tanniru, M., *A Theoretical Framework for Knowledge Transfer in Process Redesign.* The DATA BASE for Advances in Information Systems, 2008. **39**(3): p. 19.

[14] Pallent, J., *SPSS Survival Manual. A step-byStep Guide to Data Analysis Using SPSS for Windows (Version 15)*. 3rd edition ed. 2007: Open University Press.