

# Interactive Animation and Affective Teaching and Learning in Programming Courses



Alvin Prasad and Kaylash Chaudhary

**Abstract** Programming is an essential subject in computer science and information systems developments, and its demand in the computer industry is even higher today. To the dismay of the industry, there is a very high dropout rate of programming students as they find coding difficult. The literature presents many ways to combat attrition rate in programming courses, which include the way programming is taught (teaching) and received (learning) by students. This paper will focus on teaching and learning aspects of programming courses. In particular, we discuss an interactive tool that can be used to teach and learn to program. This tool is developed using Adobe Flash CS5, specifically the Adobe Action Script 3.0, Adobe AIR and iPhone OS. The tutorials are animated and designed using natural language.

**Keywords** Animation · Affective teaching · Home language · Programming

## 1 Introduction

The article by CBC News published in 2016 states that there will be 200,000 communication and information technology jobs that will need filling in Canada, and they predict that there will not be enough people to fill them [2]. According to US bureau of labour statistics employment in the area of information technology is expected to increase by 13% by 2026 [12]. In another survey, across 43 countries and territories reports that IT skills are in demand and position, which is very difficult to fill, where computer programming is one of them [3]. Students do not want to take programming courses as they feel that programming is difficult to understand, which leads to a shortage of skilled workers in this specialisation.

---

A. Prasad (✉)  
The University of Fiji, Lautoka, Fiji  
e-mail: [alvinp@unifiji.ac.fj](mailto:alvinp@unifiji.ac.fj)

K. Chaudhary  
The University of the South Pacific, Suva, Fiji  
e-mail: [kaylash.chaudhary@usp.ac.fj](mailto:kaylash.chaudhary@usp.ac.fj)

Furthermore, the proposed study's goal is to motivate students and change their attitude towards programming. The finding from this research will benefit the stakeholders such as curriculum developers, course facilitators and other researchers in making an efficient and practical decision. A system was developed to teach and learn programming where the essential concepts from the literature are incorporated. The research intends to find whether degree programmes developed in programming students mother tongue helps make students grasp the subject of programming better or not. Thus, the research question for this paper is to determine the relationship between independent variables such as interactive animation, effective teaching and learning, with the dependent variable student results in a programming course.

The next section will discuss the challenges faced in teaching and learning programming. Section 3 will present the methodology of this research, while Sect. 4 discusses the application and its architecture. This paper concludes in Sect. 5.

## 2 Challenges Faced in Teaching and Learning Programming

Some of the problems in teaching and learning of programming language are related to the student's prior knowledge and learning environment before taking the course. According to Oroma et al., the background of the student, such as the type of skill acquired from the previous training or study environment, is an important factor in the selection of subject at a tertiary institute [8]. Secondly, the lack of encouragement, and motivation and painting a negative picture in the minds of prospective students that programming is a tricky subject is another issue that needs to be dealt with by the instructors and parents. Also, mathematics is another subject which discourages students from doing computer programming because students find mathematics difficult and have stereotype ideas that programming is similar to mathematics, and hence, they refrain from choosing computing science as a major.

Furthermore, the medium or the mode of study is another factor that leads to students not liking the subject. Different learners have different taste, so a particular method might appeal to some and not to the others. Together with the medium, the teacher also plays a leading role in a student's life for further studies. Hence, a teacher's negative attitude towards the subject also demoralises the student [7]. Language is also a factor that leads to students understanding of the subject matter. The researchers in a study concluded that students whose first language is not the same as that used in the instruction of the course did not perform well [9].

According to a study Krpan et al. and Woei et al., the schooling and learning computer programming is perplexing [6, 13]. Our understanding is if someone gets to hear the statement that 'programming is difficult' every day, automatically it will become difficult for that person. Individuals need to understand that programming is just another language for programmers to communicate with computers. Information technologies personnel facilitated by computers are trying to make every task easy,

solve most of our problems and help in enhancing the world to the new technological era. We can imagine that in the coming years, computers will drive the world, and we need drivers who can authorise or command. So to take this commanding position, we should know the language which is necessary to do this. According to Jacobsen, experts during a code week discussed the need for programmers with the ability to solve problems [5]. They are predicting an increase in the demand for programmers.

Soon, education will comprise virtual learning [11], mobile learning [1], gamification with digitalised classrooms and digital books [4]. Many applications, various techniques and different methods have been tested to encourage students to take programming courses. Live coding, virtual learning environments, massive open online courses, 3-dimensional folktales, online trainer and judges, visualisation, robotic toolkits, games, mobile learning, social media and other methods have been employed to promote the interest in programming. Article [10] discusses these techniques and elaborates on some drawbacks which we want to overcome in the developed application.

### 3 Methodology

Firstly, this research reviewed the innovative technologies and methods which are currently used in teaching and learning programming through a critical literature review. Secondly, animated applications have been developed using Adobe Flash CS5, precisely the Adobe Action Script 3.0, Adobe AIR and iPhone OS document. The animation contains commands in English (medium for teaching in schools and tertiary institutes in Fiji), Hindi (Fiji Hindi spoke mostly by Fijians of Indian Descent) and I-Taukei (spoken by indigenous Fijians). In this research, an experiment was conducted using this animation and application with first-year information technology students in the Western Division of Fiji.

### 4 Application Architecture

The application used in this study as a teaching resource was developed using Adobe Flash CS5. We chose this platform because it allowed us to develop animations quickly and played on any platform with a flash player. The Adobe AIR 2 and iPhone OS within Adobe Flash were used to develop the application used on desktop, web and mobile devices. Action script 3.0 was used for the coding of the application. iPhone OS document in Adobe allows to create or develop applications deployed on Apple iPhone and iPod touch. Adobe AIR 2 document enables to create applications and bundle the same code into native apps for Windows, Mac, iPhone, iPad, Kindle Fire, Nook Tablet and other android devices. It enabled us to create an interactive environment using different controls. Majority of the students in the new era have a mobile device; the application developed can be used by the students on the go as

well. As the students are travelling to the university or anywhere, sitting in the park, on the bed trying to sleep, or anytime they want to, they can go through the animation as this will be a kind of entertainment for them, and at the same time, they will learn programming.

The animation includes interactivity, where students were able to move the mouse cursor on the code to find what a particular syntax signifies, as shown in Fig. 1. Together with that, drag and drop animations can be used to test knowledge after the explanation of the method or concept. Students can use the animation to go through while doing tutorial and lab questions.

In addition to this, the main application makes use of scenarios related to student’s daily lives, such as the characters in a village setting asked to collect mangoes from the nearby farm. As the learner starts with the first challenge, other characters with new and advanced methods are introduced. Every challenge adds a new line of code to the existing block of code, which leads to a complete program. Figure 2 shows the flow of the challenges in the application.

Figure 3 shows an example scenario. The learner gets the first challenge that is to create a basket by writing code, as shown in Fig. 4. We believe that this challenge will motivate the learners whereby they would like to overcome the contest. It will

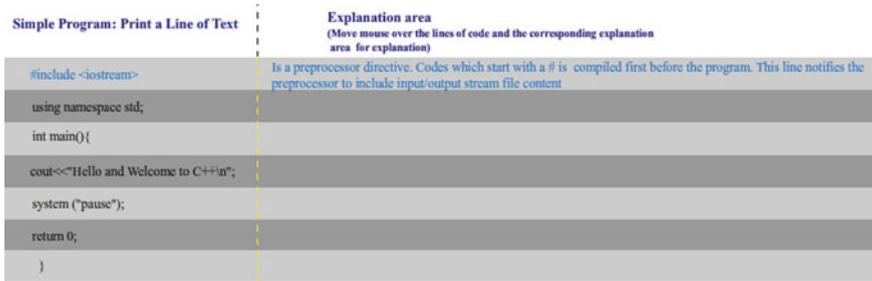


Fig. 1 Explanation of individual lines of code

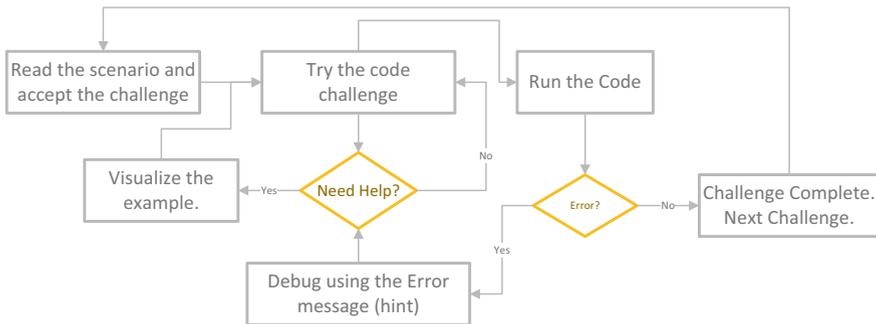


Fig. 2 Flow chart shows how the application works

**Fig. 3** Initial page with a scenario



lead to the application and understanding of different concepts in programming. If, for instance, a learner needs help, the application has example pages which explain what steps to take. Figures 5, 6, 7 and 8 show example pages from the app. The example can be described using any three languages. Students have the option of going to the example page over and over, which explains the concept in an animated format. After writing the correct code, the learner will be able to see their creation. It also will provide an option which will lead to the next challenge. If the code entered is incorrect, then an error message will appear with an explanation on the error and hint which will help in better understanding of the challenge.

While completing the challenges, learners will ultimately write blocks of code, which eventually will solve a problem. As shown in Fig. 9, learners after completing five challenges can write a block of code where they can declare variables, initialise the values and write the formula to add the number of mangoes. While they write the code, the animation continues to show what they have done. After completing these codes, learners will be able to use the concept learnt to do any calculations like addition, subtraction, multiplication and division on any number of variables.

## 5 Result

A total of 68 students result was recorded before the use of animation in teaching and post to the use of animation. Exam results were recorded, and statistical analysis was carried out. As the sample size and the students are the same for both the exams, a paired sample t-test was conducted to determine the effect of animation on the learning ability of students. Some assumptions have to be satisfied to conduct this test. One being that the dependent variable (exam results) is continuous; two, the independent variable (students) must comprise related groups; three, there ought to be no significant outliers in the differences between the two associated groups, and lastly, the distribution of the differences in the dependent variable between the two related groups should be approximately normally distributed.

The data in sample satisfies the first and second conditions, that is, the dependent variable—exam results are measured on a continuous scale—and the independent

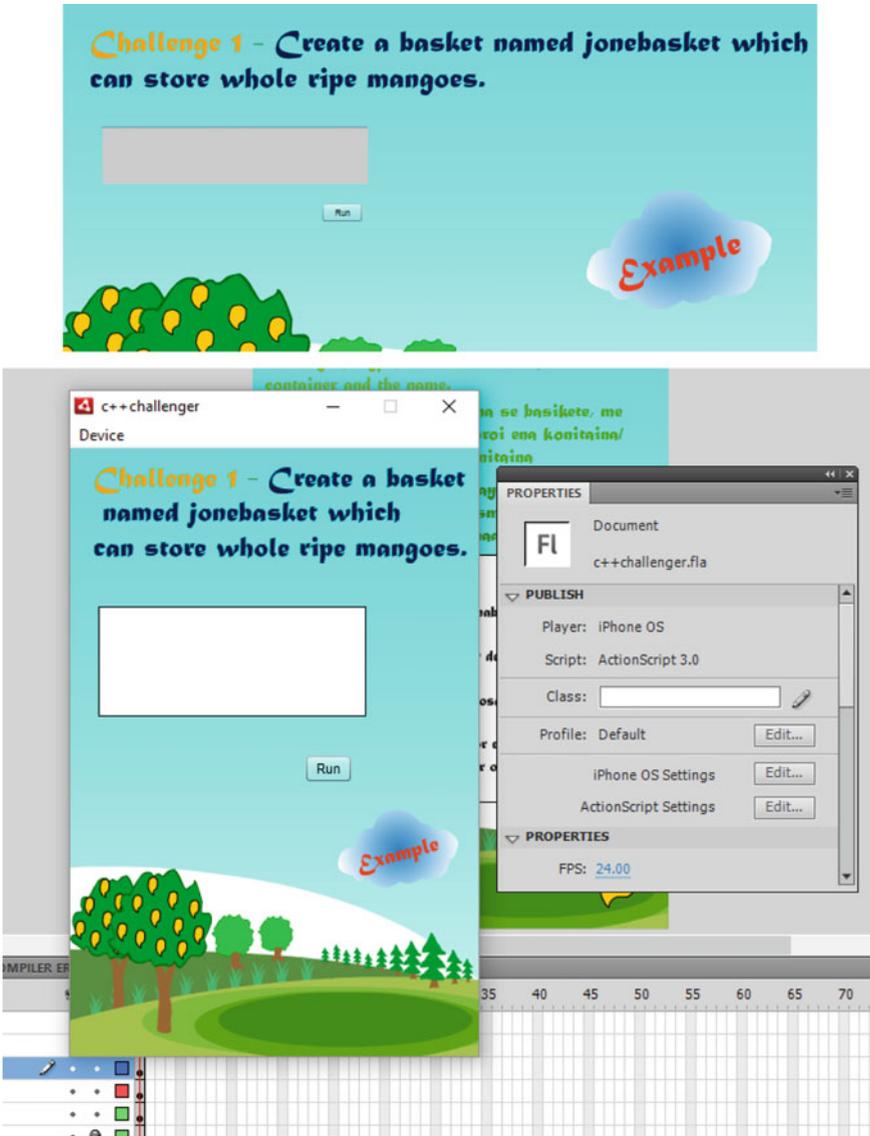


Fig. 4 Challenge 1 writes the code to create a basket

variable—students contain the same students who have set for the exams. For the third assumption, a specific procedure needs to be carried out to determine if there are any outliers in the data. To proceed, find the differences between the two results and divide into quartiles. We shared the data into three quartiles ( $Q_1$ ,  $Q_2$  and  $Q_3$ ) by using the quartile function from excel and later found the inter-quartile range (IQR)

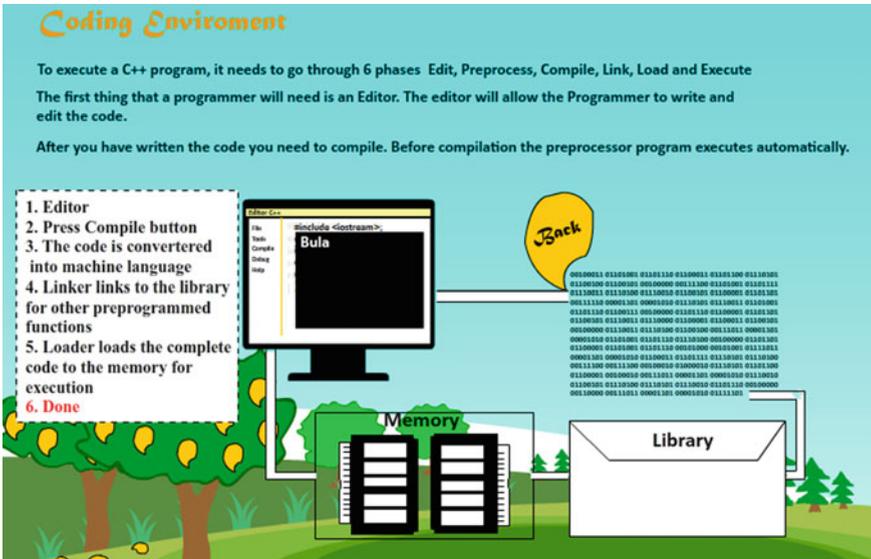


Fig. 5 Animation showing the coding environment



Fig. 6 Error message with hints

by subtracting the first quartile from the third quartile. Then, the upper bound and the lower bound are found by using the following formulas:

$$\text{The formula for Upper Bound: } Q3 + (1.5 * IQR) \tag{1}$$

$$\text{The formula for Lower Bound: } Q1 - (1.5 * IQR) \tag{2}$$

The first quartile ( $Q1$ ) is four, while the third quartile ( $Q3$ ) is seven, as shown in Table 1. The upper bound is 11.5, and the lower bound is  $-0.5$ . To satisfy the third assumption, that there should not be any significant outliers. According to the data obtained after conducting the outliers test, there are no noteworthy outliers in the data; thus, so far, three assumptions have been proven true for 'sample' data.

**To create a variable or a container one needs to identify the type(data to be stored) of variable/ container and the name.**

**Me buli e dua na konitaina se basikete, me kilai kina na data me marovoi ena konitaina/basikete, koi na gaca ni konitaina**

**Variable ya container banage ke wastin important hai ki jano ki usme kon type ke chiz rakhega, aur container ke naam konchi dega**

**Data types**

int -> +/- whole numbers or +/- maha taucoko mela na eg. 6, -2,100,-215

double -> +/- decimal numbers or desimolo eg. 3.2, 1.55, -45.2323

string -> words/sentences or na vosn/yatu vosn eg. "hello"

char -> single character eg. 'S'

bool -> represents 1 or 2 states or e vakaraitaka e 1 se 2 nai tuvaki like true or false/ on or off or dina/lasu

**double y = 3.14;**

y = 3.14

Container named y is created which can store positive and negative decimal numbers and stores a value of 3.14

Back

Fig. 7 Example page in three languages

**Woow.....Challenge Complete!**

**Jone's basket is ready.**

Next Challenge

**Woow.....Challenge Complete!**

**A basket named total is created.**

**Lets help them get the total mangoes collected in the next challenge.**

Next Challenge

Fig. 8 Two different challenges completed page

Fig. 9 Simple coding completed after completing five challenges

**Challenge 5 - Lets add the mangoes from jone basket and mere basket and store the total amount in the basket named total.**

```
int jonebasket = 11;
int merebasket = 7;
int total = 0;
total = jonebasket + merebasket;
```

Hint Run

Example

**Table 1** Finding outliers for sample

First quartile ( $Q1$ )	4
Second quartile ( $Q2$ )	6
Third quartile ( $Q3$ )	7
Inter-quartile range (IQR)	3
Upper bound	11.5
Lower bound	-0.5
First value outlier	FALSE

Furthermore, the data used for paired sample t-test should be approximately normally distributed. To test for this, use a histogram, but a probability plot is paramount for determining the distribution. Regression analysis in excel identified normality checks. It creates a normal probability plot displaying the  $R^2$  value, and a linear trendline is added to indicate the deviation of the difference values from this trendline. In Fig. 10, the trendline (theoretical percentiles of the normal distribution) is the thin black line while the blue dots (observed sample percentiles) are the plots for the differences. It indicates that the data is almost normally distributed with a few insignificant outliers. The  $R^2$  value, 0.86, is closer to 1, demonstrating that data closely fit the regression line; thus, it is approximately normally distributed.

We can now conduct a paired sample t-test since the data in sample satisfies the four requirements mentioned earlier, as shown in Table 2.

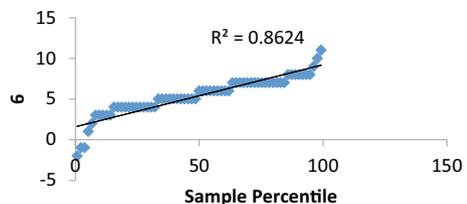
The null and alternative hypothesis of the research is:

Ho There is no significant difference in the Pre\_Score and Post\_Score means of students.

Ha There is a significant difference in the Pre\_Score and Post\_Score means of students

Since the  $p$ -value is less than the significance level ( $0 < 0.05$ ), reject the null hypothesis. That is, there is a significant difference in the Pre\_Score and Post\_Score means of students. Thus, it concludes that the use of animation allows students to understand the concepts in programming.

**Fig. 10** Normal probability plot for sample



**Table 2** t-test: paired two sample for means

	Pre_score	Post_score
Mean	4.985,294,118	10.39705882
Variance	6.19381036	6.60118525
Observations	68	68
Pearson correlation	0.561131223	
Hypothesised mean difference	0	
Df	67	
<i>t</i> Stat	-18.82631138	
$P(T \leq t)$ one-tail	9.28691E-29	
<i>t</i> critical one-tail	1.667916114	
$P(T \leq t)$ two-tail	1.85738E-28	
<i>t</i> critical two-tail	1.996008354	

## 6 Conclusion

Almost every day, new technology is invented, and its use improves the lives of billions of people. Moreover, very few students are interested in pursuing undergraduate studies in programming as they feel that it is very difficult to understand. Thus to encourage more students to undertake this subject area, an animation was developed, which teaches fundamental concepts in programming. We tested a group of first-year programming students' results before learning via this animated concept and after learning programming incorporated animation. The result indicates that there is a significant relationship between the use of animated tools and academic performance. The result of this study will help students, course facilitators and researchers in knowing how feasible are animation and effective teaching and learning in regard to teaching difficult subjects such as computer programming.

### Compliance with Ethical Standards

**Conflict of Interest** The authors declare that they have no conflict of interests.

**Ethical Approval** This research concerns students. The exam score of students was used in the study as per their ethical approval.

**Informed Consent** Informed consent was obtained from all individual participants included in the study.

## References

1. M. Ally, J. Prieto-Blázquez, What is the future of mobile learning in education? *Univ. Knowl. Soc. J.* 142–151 (2014)
2. CBC News, Why learning Computer Coding is so important, according to advocates (20 Oct 2016)
3. A. DeNisco, Report: 40% of employers worldwide face talent shortages, driven by IT. *TechRepublic* (2016)
4. J. Hamari, K. Jonna, H. Sarsa, Does Gamification Work?—a literature review of empirical studies on gamification, in *47th Hawaii International Conference on System Science* (IEEE computer society, 2014), pp. 3025–3034
5. H. Jacobsen, Digital experts say coding leads to empowerment (and jobs). *EurActiv.com* (2015)
6. D. Krpan, S. Mladenović, M. Rosić, Undergraduate Programming Courses, Students' Perception and Success, in *International Conference on New Horizons in Education. Procedia—Social and Behavioral Sciences*, pp. 3868–3872 (2015)
7. V.P. Meisalo, S. Suhonen, S. Torvinen, E. Sutinen, Formative evaluation scheme for a web-based course design, in *Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education* (Association for Computing Machinery, New York), pp. 130–134
8. J.O. Oroma, H. Wanga, F. Ngumbuke, Challenges of teaching and learning computer programming in developing countries: Lessons from Tumaini University, in *Proceedings of INTED Conference* (Spain, 2015), pp. 3820–3826
9. N. Pillay, V.R. Jugoo, An investigation into student characteristics affecting novice programming performance. *ACM SIGCSE Bull.* 107–110 (2005)
10. A. Prasad, M. Farik, Integration of innovative technologies and affective teaching & learning in programming courses. *Int. J. Sci. Technol. Res.* 313–317 (2015)
11. L.G. Rolando, D.F. Salvador, A.H. Souza, M.R. Luz, Learning with their peers: Using a virtual learning community to improve an in-service Biology teacher education program in Brazil. *Elsevier* (2014)
12. U.S. Bureau of Labor Statistics, Occupational outlook handbook. Retrieved March 6, 2019, from U.S. Bureau of Labor Statistics. <https://www.bls.gov/ooh/computer-and-information-technology/home.htm> (13 Apr 2018)
13. L.S. Woei, I.H. Othman, C.K. Man, Learning Programming using Objects-First Approach Through Folktales (Penerbit UTM Press, 2014), pp. 47–53