

Using Visual Logic with Pseudocode to Teach an Introductory Programming Course

G. Cooper, PhD

Engineering and Information Sciences, DeVry University, Downers Grove, IL, USA

Abstract - *Introductory programming courses experience large attrition rates due to the challenges of the material and incoming student learning styles. CIS115 is an introductory logic and design course at DeVry University. To improve student satisfaction and reduce attrition in the course, Visual Logic (www.visuallogic.org) with pseudocode was introduced as an alternative to a standard programming language. Using Visual Logic, students were able to create executable flowcharts while learning fundamental programming concepts. Pseudocode was taught in conjunction with Visual Logic so that students learned to translate the flowchart into workable code. Visual Logic was employed in both an online and onsite delivery of the course. Student satisfaction survey data and grades were evaluated. Grades from this course were compared with the course when a high level programming language was taught. This paper explains the outcomes of using Visual Logic with pseudocode in an introductory programming course and explores the advantages and disadvantages of this approach.*

Keywords: Visual Programming, Programming concepts, Introductory Course, Logic and Design, Computer Science

1 Introduction

Introductory programming courses are faced with several challenges. Students enter the course with varying degrees of prior knowledge and comfort with programming. CIS115 is a Logic and Design course at DeVry University. The course introduces program design and basic logic with pseudocode and flowcharts. No programming languages are used in the course; rather all assignments use Visual Logic and pseudocode.

Visual Logic is an executable flowchart program utilizing a graphical interface to illustrate programming topics. The flowchart provides users with the ability to create loops, decisions, arrays, procedures, and files. Users execute the flowchart to produce results and can interact with the flowchart using basic input/output commands. Visual Logic removes the frustration of syntax errors and new integrated development environments (IDEs) from novice programmers. The challenge is in creating the logic rather than managing a new development environment.

Students use Visual Logic in the first programming course they take. The course is delivered in both an online and onsite format with the software available on a remote Citrix site. Students login to the site, create flowcharts to implement their assignments and store the Visual Logic files on the remote site.

Research has shown that students can learn to program effectively using graphical tools and non-standard programming environments [6][7][8]. Multiple graphical-based program teaching tools have been developed and implemented at universities. Most notably, Alice, developed by Carnegie Mellon in 1999 [3] provides new programmers with an object oriented approach to programming. Alice offers students the ability to create 3D animations while learning fundamental programming topics. RAPTOR, created by the Air Force Academy in 2005 [2] uses a flow model environment to teach programming concepts. In a study of three final exam questions given to students with the option of using traditional programming languages or RAPTOR, students choosing RAPTOR to complete the final exam questions scored significantly higher than those using traditional programming languages [2].

In addition to these results, in 2011 Gudmendsen et al. completed a study using Visual Logic in three different approaches to determine its effectiveness in the classroom[5]. The first course used Visual Logic to introduce algorithms in a general education computer science course; the second course used Visual Logic for six weeks then transitioned to Python. The third course used Visual Logic in conjunction with teaching Java. In all three courses using Visual Logic enhanced student learning.

This paper takes a new approach by introducing Visual Logic in conjunction with pseudocode. The pseudocode is presented as equal in importance to Visual Logic in learning logic. To determine the effectiveness of Visual Logic with pseudocode in CIS115, two areas were examined. End of course student surveys and grades for four sessions of CIS115 was gathered. This data was compared with four sessions of the course when a high level programming language was taught. The question this research sought to answer was "do more students understand the material, enjoy the course, and pass the course when Visual Logic is taught as compared to a high level

programming language?" This paper describes the findings and extrapolates on methods to improve the course.

2 Initial Course Design

The Logic and Design course is one of the first courses that new freshmen take in their major. This course is taught in a lab/lecture classroom in which each student has a computer at his or her desk. In addition to this format, the course is also offered online. CIS115 is a 3 credit hour class and the course description is below:

This course introduces basic programming logic, as well as algorithm design and development, including constants, variables, expressions, arrays, files and control structures for sequential, iterative and decision processing. Students learn to design and document program specifications using tools such as flowcharts, structure charts and pseudocode. Program specification validation through desk-checking and walk-throughs is also covered.

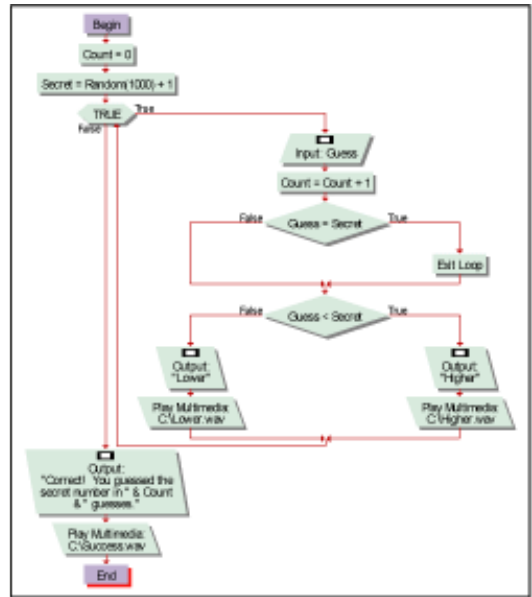
To assist students with the transition from logic and design to programming, pseudocode is an important part of this course. Students must create pseudocode that matches the Visual Logic program. Prior to introducing Visual Logic, new programming students used C++. The same topics were covered in the course regardless of the teaching tool.

The maximum allowed number of pages is seven for Regular Research Papers (RRP) and Regular Research Reports (RRR); four for Short Research Papers (SRP); and two for Posters (PST).

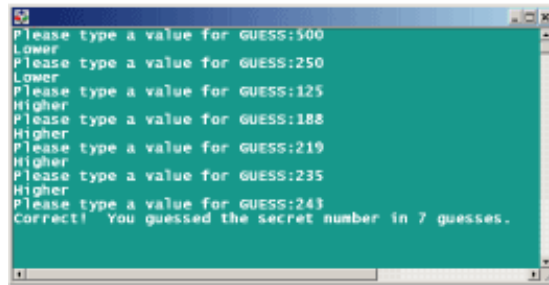
3 Visual Logic

Visual Logic provides students with a visual method to create programs. All assignments require students to create an IPO (input, processing, output) chart, a Visual Logic flowchart, and pseudocode.

Visual Logic (visuallogic.org) is a visual flowchart program available for download as a trial version [9]. The program can be purchased by universities or individuals. Visual Logic will only run on Windows operating systems, which presents a problem for students using other operating systems. The university has a centralized server for students to access using a Citrix client. Visual Logic is available for all students with a login to the site. Figure 1 shows an example of a Visual Logic screenshot.



Visual Logic Flowchart solution to the High-Low Game



Visual Logic flowchart execution output

Figure 1: Visual Logic flowchart and output

Using Visual Logic, students can create complex programs with multiple programming structures. Visual Logic has no specific data types so data types are implemented in the pseudocode. Even so, Visual Logic supports arrays, if statements, pre and post-test loops, text file I/O, logical operators, and procedures.

Students create complex flowcharts with Visual Logic and learn basic logic problem solving. They can run their flowcharts to determine the accuracy of their design and give themselves a sense of accomplishment.

4 Visual Logic with Pseudocode Implementation

Visual Logic provides the opportunity for students to witness their programs running without the usual syntax errors that come from learning a new programming language. Each assignment is composed of an IPO chart, pseudocode, and a Visual Logic flowchart. The IPO chart is a high level overview of the problem the students are solving. For

example, if the problem is to calculate the area of a circle after the user enters the radius, the input would be the radius, output is the area of the circle and the processing is $\text{area} = 3.14 * \text{radius} * \text{radius}$.

The students write pseudocode with basic syntax. The students use keywords such as: START, DECLARE, PROMPT, INPUT, DISPLAY, CALCULATE, SET, IF/THEN/ELSE, DO/WHILE, FOR, WHILE, SWITCH, STOP. Since pseudocode is language-independent, only three data types are discussed in the class. These data types are real, integer, and string.

The following is an example of pseudocode to calculate the area of a circle after the user inputs the radius.

START

DECLARE Real radius.

DISPLAY "Enter radius: "

INPUT radius

DECLARE Real AreaOfCircle

SET AreaOfCircle = $3.14159 * \text{radius} * \text{radius}$

DISPLAY "The area is: "+AreaOfCircle

STOP

Relational operators and Boolean logic are covered with selection and repetition structures. The pseudocode is stressed to the students as one of the most important parts of the course. Pseudocode translates nicely to high level programming languages.

The assignments account for 52% of the total course grade. The remaining points are distributed between threaded discussions, quizzes, and a comprehensive final exam. At the end of the course students have a firm foundation in logic and design and are well prepared to start programming in a high level language.

Visual Logic has several advantages for beginning programmers. The main benefit is the visual nature of the program. Students can visualize input, output, processing, decisions, and repetition in their flowchart. They can create large flowcharts with relative ease and run their programs to see them working. Most students are comfortable working with Visual Logic after one short tutorial. Another advantage of Visual Logic is that it can be used with only a web browser and internet connection. Students login to the Citrix client and use Visual Logic from there. Some students choose to download the trial version, but most access the program using Citrix. Since CIS115 is taught in both the online and onsite modality the ease with which the students can access the software is an important factor.

Writing pseudocode using a word processor has the advantage that students can focus on logic instead of coding syntax. Many IDEs have their own learning curves whereas pseudocode can be written using a word processor.

While Visual Logic has many benefits over traditional programming to introduce new programmers to logic and design, students fear they will not use Visual Logic again. They are eager to start programming and use their new skills with a high level programming language. Visual Logic is basically a teaching tool and they may not encounter this program in industry.

5 Results and Discussion

Two areas of student achievement were studied to determine the success of Visual Logic with pseudocode in CIS115. These areas were end of course student surveys and grades. Pass rates in introductory programming courses are notably low [1]. Bennedsen and Caspersen studied institutions around the world and the pass rate for a computer science 1 course. The average pass rate in these courses is 67%.

Each session, online surveys are administered to students. These surveys are optional, yet students are encouraged to complete them to give the instructor feedback on the course. The survey consists of six main areas of feedback and a comment section. The composite score of the course along with the comments were evaluated. Students rated the course from 1 to 10 with 10 being the highest score.

The mean student satisfaction with the course was 8.57 for all students. For the onsite course the mean satisfaction was 9.75 and the online student satisfaction was 8.42. Four onsite students participated in the survey and 32 online students participated. It is interesting to note that onsite students had approximately 13.7% higher satisfaction with the course than online students. Table 1 illustrates the number of students who responded with scores from 1 to 10.

These results are noteworthy since student satisfaction can lead to more motivated learners, higher grades, and better persistence. Over half of the students rated CIS115 with a 10, the highest score and 72% of students rated CIS115 with a 9 or 10. This indicates a high degree of satisfaction with the course overall. This student satisfaction survey was not administered to students when the course was delivered with C++.

Table 1. Student Satisfaction Results based on Survey Data

Course Score	Number of student respondents
10	20
9	6
8	4
7	1
6	1
5	2
4	2
3	0
2	0
1	0

In addition to student satisfaction surveys, student grades were assessed. The number of students who achieved grades of A-C, and those who passed and failed were tabulated for four sessions of CIS115 and are illustrated below in Table 2. This data does not include students who dropped the course, only those who completed the course and took the final exam.

Table 2: Student Grades for Four Sessions of CIS115

	Percent Grades A-C	Percent grade D	Percent Passed (grades A-D)	Percent Failed
Before Visual Logic	78.57%	12.5%	91.07%	8.93%
After Visual Logic	76.93%	10.68%	87.38%	12.62%

While the pass rates of both implementations of the course are high, more students passed using a high level programming language. The pass rate for C++ is approximately 4% higher than the pass rate for Visual Logic. While the students enjoy CIS115 and about 3 out of 4 students had a reasonable understanding of the subject material, more students passed the course when C++ was used rather than Visual Logic. While Visual Logic was intended to improve the pass rate, it actually decreased the pass rate slightly.

The results were surprising but can be explained when further examined. One of the main reasons to use Visual Logic is to reduce frustration from learning syntax and IDEs. Microsoft Visual Studio, the IDE used when programming C++ by Devry University students is user friendly as compared to other IDEs or using text editors.

Another advantage of using C++ is that students will use this language in their careers whereas Visual Logic is basically a training tool. Many students complained that they would never use this program again. Students who start immediately in a programming language can work through problems in an actual IDE and test their algorithms with code. Students may be more motivated to learn the material when they are learning a programming language that they can expect to use in their careers as opposed to Visual Logic which they do not expect to ever see outside of an academic environment. They can find satisfaction in using an actual language and creating programs in that language.

6 Course Improvement

The results from this study indicate that while students are satisfied using Visual Logic with pseudocode in their introductory programming course, they are not as successful as using a high level programming language. These results are noteworthy, and are counter to the results found from other researchers [2][5]. This course can still be improved to increase student comprehension and reduce frustration with syntax errors.

Writing pseudocode prior creating programs is a good habit for the students to learn. Instead of combining pseudocode with Visual Logic, pseudocode could be combined with C++ or Visual Basic. Visual Basic is a user friendly programming language with relatively easy syntax to understand.

Another improvement to the course would be to introduce a scripting language to coincide with Visual Logic and the pseudocode. A scripting language such as python would be easier to learn and a good marketable skill for the students.

An area of further study would be determining the extent to which the skills learned in CIS115 transfer to the subsequent course. Student survey data and grades from the next course could be analyzed to ascertain their satisfaction with the course and mastery of the subject material.

These options for course improvement provide opportunities for students to test the knowledge they gained from CIS115. A combination of Visual Logic, pseudocode, and a scripting language or Visual Basic and pseudocode can be implemented and the impact on student satisfaction, retention, and grades analyzed.

7 Conclusions

Introductory computer programming courses are challenging courses for students to take and instructors to teach. Determining the best method of explaining logic and design to new programmers while balancing the aggravation from syntax errors can be difficult. To improve the student experience Visual Logic with pseudocode was introduced in CIS115. The course was previously taught using C++. Data

was analyzed for CIS115 to determine student satisfaction, knowledge of the subject material, and pass rates. The data consisted of student satisfaction from end of course surveys, and grades. The course had an 87% pass rate with 72% of students rating the course with a 9 or 10 out of 10. However, prior to using Visual Logic the course had a 91% pass rate indicating that using Visual Logic with pseudocode had little effect on improving student retention and understanding of the material.

To challenge students and improve their comprehension of the material, adding a scripting language or Visual Basic were explored and explained in the Course Improvement section. Based on the findings of this analysis, using Visual Logic with pseudocode to introduce logic and design was not as successful as using C++ for CIS115 at DeVry University.

8 References

- [1] Bennedsen, J. and Caspersen, M. "Failure Rates in Introductory Programming." *ACM SIGCSE Bulletin*, v. 39 n.2 2007.
- [2] Carlisle, M., Wilson, T., Humphries, J. and Hadfield, S. "RAPTOR: A Visual Programming Environment for Teaching Algorithm Problem Solving." *SIGCSE '05 Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*. p 176 – 180.
- [3] Cooper, S. Dann, W., and Pausch, R. "Alice: a 3D Tool For Introductory Programming Concepts." *Journal of Computing Sciences in Colleges*, v 15, n 5. 2000 p107-116.
- [4] Crews, T., and Ziegler, U. "The Flowchart Interpreter for Introductory Programming Courses." *Proceedings of FIE '98*, 307—312
- [5] Gudmendsen, D. Olivieri, L., Sarawagi, N. "Using Visual Logic: Three different approaches in different courses – general education CS0, and CS1". *Journal of Computing Sciences in Colleges* v 26, n 6, 2011, p 23 – 29.
- [6] Kulik, C. and Kulik, J.. "Effectiveness of Computer Based Instruction: An Updated Analysis". *Computers in Human Behavior*, v. 7 issue 1-2, 1991, pp. 75-94.
- [7] Shackelford, R., and LeBlanc, R. "Introducing Computer Science Fundamentals Before Programming." *Proceedings of FIE '97*, 285—289
- [8] Wenglinsky, H. "Does it compute? The relationship between educational technology and student achievement in mathematics." *Educational Testing Service Policy Information Center*. 1998.
- [9] Visual Logic. <http://www.visuallogic.com>. 2013.