Engaging CS1 Students With Project Based Learning

Michael Cassens
Department of Computer Science
University of Montana
Missoula, USA
michael.cassens@umontana.edu

Yolanda Reimer
Department of Computer Science
University of Montana
Missoula, USA
yolanda.reimer@umontana.edu

Abstract— **This Research to Practice Work in Progress paper examines a persistent issue with many CS1 courses – finding assignments that are engaging while still covering core learning objectives necessary for success in the course and computer science programs. In traditional CS1 courses, textbook tasks are usually elementary, mathematically based, and only somewhat relevant to the student. Instead, if we allow students to design and implement their own projects, they not only master the primary objectives of the course, but they can do so in a way that is most meaningful and engaging to them. While project-based approaches to CS1 are not new, most employ predefined projects created by the instructor or found online through community-based educational resources. In this research, we leverage the fact that many CS students are double majors and allow them to create projects that are unique and applicable to these other interests. We contrast a traditionally taught CS1 class with a project-based CS1 class and compare student achievement between the two. We discover that students perform better on most learning objectives in the project-based class and they gain a deeper understanding of how to scaffold programming components into follow-on assignments. In addition, students indicate that the project-based focus was more enjoyable and more useful as they create projects that apply to their personal interests and thus their future.**

Keywords— CS1; Project-based assignments

I. INTRODUCTION

Students often bemoan the fact that standard textbook assignments are too narrow and prescriptive, and that they require no creative input. Previous studies examine different approaches to teaching CS1/2 courses, including object-first to object-last approach [1], gamification and relevancy-based assignments [2-7]. Others integrate novel changes to introductory courses, such as focusing on creativity without having to write much code. In particular, Guzdial's research focuses on attracting students who might be interested in computational studies but who do not want to become coders, and he has seen a great deal of success with his methodology [8,9]. While some of these methods have proven successful when compared to traditional assignments, many are still more instructor driven than student driven [10]. Additionally, retaining students after they have chosen Computer Science as a major remains a challenge. Even with game-based or creative assignments, while some students may find them compelling, others do not. Given the variety of different backgrounds and interests students come from, using a stock approach to assignments simply can not appeal equally to all. As a result, students often lose interest in the traditional class material as they feel it does not relate to them, and they may choose to drop out of Computer Science altogether.

To alleviate some of these issues, we experimented with integrating project-based learning into a CS1 course and comparing it with a traditionally taught CS1 course across specific student learning outcomes. There is a great deal of research to indicate project-based assignments not only help students retain information, but they also improve motivation across disciplines and age groups [11-17]. Instructors of project-based learning feel more engaged in their work as well [18].

To construct our study, we first administered a short online survey to gain a better understanding of the students in each of the two courses. Of the twenty students who responded, 45% were either computer science majors or seriously considering computer science as a major. The remaining 55% were majoring in Physics, Anthropology, or Management Information Systems. Students indicated that they were interested in learning new problem-solving skills, helping others, and increasing their skillset in their field. Particular problems that students wanted to solve included isolating electrons in electron shells, harnessing the potential of plasma, and advancing animal awareness and cultivating their survival.

Capitalizing on these varied interests, we hoped that the project-based CS1 class would allow students to explore areas of particular relevance to them, and by doing so, they would realize first-hand the value and importance of computer science. Our approach not only gives students the ability to choose a subject matter that relates to their unique interests, but it also teaches them how to solve problems while simultaneously accomplishing core computational thinking and programming objectives.

II. METHOD

This study involved two different CS1 sections. Both courses ran concurrently throughout the same semester. Although there were two separate instructors, the book and material covered were the same across both classes. The time spent in lecture and lab was directed entirely by the instructor of each section. Although the teaching style of the instructors was

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

different, we do not believe this had a causal effect on the output as each instructor was well-versed in the material being taught. In the traditionally taught class, or control group, there were 30 students enrolled; in the project-based class, there were 19 students enrolled. All students self-enrolled into their respective section.

A. Assignments

In the project-based learning class, the instructor provided a brief overview of project requirements and outlined general concepts that students needed to cover in each assignment. As long as each of these concepts was incorporated, projects would be deemed valid and complete. Milestones were established for each project to ensure that appropriate progress was being made and that the instructor had ample opportunities to provide feedback throughout the semester. Once project proposals were finalized, students were required to design and describe implementation and user interaction details. Initially, UML diagrams were not required, but students used an outline summary to describe their project. All projects in both classes used Java as the primary programming language with the 1.8 JDK and Eclipse IDE Neon.2.

For the first homework assignment in the project-based course, students were given the following description and required learning objectives that they needed to achieve without direction; beyond that, students had the freedom to apply these objectives to their specific interest. Students had four and a half weeks to complete this project.

(Project-based class, HW 1). In this homework, you are to create your project. You get to decide what is interesting to you. To complete this assignment, you are first to create an outline that describes what you would like to accomplish. Try to be as specific as you can be. Add as much functionality as you would like. As long as you complete the following requirements below, you will have successfully fulfilled the requirements of this homework.

1. *Create an outline of your project.*
2. *Make sure the program uses proper formatting and commenting*
3. *Use different variables to store information*
    a. *Use both standard variables, constants, and proper naming conventions*
4. *Get user input in the console*
5. *Use different relational and equality operators where necessary*
6. *Use at least three different operators such as +,-,*,/, %*
7. *Make sure the calculations are separate from the printing of the result.*
8. *Make sure you keep a log of all errors that occur while creating your project.*
    a. *Describe each error (syntax, runtime or logical) in a separate document*
    b. *How did you correct each error?*

Conversely, in the traditional CS1 class, students were given the following multiple shorter homework assignments from the textbook [19]:

1. *Write a program Info.java that prints, on separate lines, your name, your birthday, your hobbies, your favorite book, and your favorite movie. Label each piece of information in the output.*
2. *Write a program Tree.java that prints the outline of a tree using an asterisk (*) characters.*
3. *Write a program that prints the phrase Knowledge is Power:*
    a. *on one line*
    b. *on three lines, one word per line, with the words centered relative to each other*
4. *Write a program Initials.java that displays your initials in large block letters, Make each large letter out of the corresponding regular character. See example in text book page 56.*

B. Learning Objectives

Table 1 categorizes the first six of twenty specific learning objectives both courses sought to achieve along with a mapping of assignments in each course that measure each objective. For example, the first learning objective focuses on students' ability to appropriately outline their problems, and is measured via Homework 1 from the project-based class, and Homework 1-4 in the traditionally taught class.

TABLE 1. Mapping of CS1 Learning Objectives

Learning Objective	***Project Based***	***Control Group***
a. Outline	HW 1	HW 1-4
b. Commenting	HW 1	HW 1-4
c. Structure	HW 1	HW 1-4
d. print and println	HW 1	HW 1-4
e. Variables	HW 1	HW 1-4
f. User Input	HW 1	HW 1-4

C. Assessment

All assignments in both classes were measured using the same rubrics and evaluated by a single graduate teaching assistant to provide consistency. Based on rubrics, students were given a score of either Beginning (1 point), Developing (2 points), Accomplished (3 points), or Exemplary (4 points) for their achievement of each learning objective. A rubric from the first project-based homework assignment measures six learning objectives as shown in Table 2.

TABLE 2. Project-Based Homework #1 Grading Rubric

Learning Objective	*Beginning (1)*	*Developing (2)*	*Accomplished (3)*	*Exemplary (4)*
a. Outline of Project	Incomplete sentences or one word answer	A single short sentence describing the project	Multiple sentences describing the project	A story describing the project including the motivation behind the project and what they hope to accomplish
b. Commenting	Comments at the top of the program and none in the program	Comments at the top of the program and few comments throughout the program	Comments at the top of the program and well commented throughout the program	Comments at the top of the program, self-commenting code and well commented where necessary
c. Structure of Program	The syntax of the program is correct	The syntax is correct, and the program is formatted well.	The syntax is correct, the code is formatted, and the code is structured with variables at the top and methods and code below	The syntax is correct; the code is formatted consistently, variables at the top and consistent naming used (i.e., Camel Case, Pascal case, etc.)
d. Use of println and print including escape sequences and concatenation	Used all println	Used some println and some print	Used println, print, and used escape sequences	Used println, print, escape sequences and concatenation.
e. Usage of variables	Created some variables	Created variables and reused them throughout their program	Created variables, used proper naming conventions, reused them	Created variables, constants, used proper naming conventions, reuse and avoid magic numbers.
f. Get user input	Asked for user input	Asked for user input and used it later in the program by printing it out	Asked for user input and used it in a calculation or used it in a concatenation	Asked user input, used it to make a choice.

III. Project Examples

Given the freedom to choose their own problems in the project-based class, students produced many interesting and highly personalized projects. This section briefly describes two such projects to give the reader a sense of scope and focus.

Sample project #1 (as summarized by the student):

The following program is designed so that the user chooses an element from the Periodic Table and then the Atomic Number, Symbol, Name, and Atomic Mass are shown. Afterwards, the user can derive the density based off of a given volume, the volume (L) from a given a density (g/L), the number of moles from a given mass (g), or grams of an element from a given mole amount.

Sample project #2:

The following program will simulate a trivia program. The goal is to create an application that allows users to test their knowledge against a set number of questions. Initially, these questions will be derived directly in the program, but as the program progresses, the user will be able to choose from a number of different categorical questions read in from a file.

IV. Results and Discussion

A. Learning Objectives

Table 3 shows the results of student achievement across our stated learning objectives for both the project-based group and the control group. Average scores for each learning objective were calculated based on rubrics like that previously shown. Results for each learning objective were tested for statistical significance using the F-Test for equal or unequal variance and a t-test for statistical significance with 95% confidence.

As indicated in Table 3, the project-based group was able to understand better and achieve the overall learning objectives, particularly early in the semester, when compared to their peers in the control group. Of the twenty learning objectives we evaluated, performance was better on seventeen of them in the project-based group as compared to the control group, and eleven of these differences are statistically significant. These results imply that the project-based group was better able to conceptualize and apply the theoretical knowledge given to them and put it into practical terms. These results also suggest that the project-based group gained a stronger foundation in fundamental concepts, which may lead to increased retention and more sustained engagement in CS over the long term.

The control group outperformed the project-based group for the *Custom classes, Class Relationships,* and *Method Overload* learning objectives. However, these results were not statistically significant indicating that this may need further study with a follow-on experiment.

TABLE 3. Mapping of CS1 Learning Objectives

Learning Objective	***Project Based***	***Control Group***	***p < .05***
a. Outline	3.579	1.767	.000015
b. Commenting	3.105	2.064	.0254
c. Structure	3.526	2.500	.0065
d. Print and println	1.526	1.333	.2897
e. Variables	3.579	2.633	.0149
f. User Input	3.579	2.667	.0344
g. UML diagram	2.000	0.833	.0024
h. Random class	2.632	2.267	.2082
i. Math class	2.631	2.400	.3037
j. Formatting classes	2.579	1.333	.0048
k. Graphic classes	3.211	0.833	.000002
l. Custom classes	2.789	2.967	.3580
m. Description in JPanel	3.211	2.000	.0031
n. Class relationships	2.368	2.400	.4676
o. Control statements	3.316	2.600	.04991
p. Loops	3.158	2.733	.1726
q. File Read	2.632	1.967	.07001
r. Advanced graphics	3.263	1.500	.000002
s. Pass by ref/val	2.316	1.767	.07564
t. Method overload	1.842	1.900	.4457

B. Scope

One of the problems that the instructor in the project-based course faced was that some projects ended up being too simplistic while others were overly complicated. Although not completely eliminated, scope issues were alleviated by reassessing project progress on a weekly basis.

C. Evaluation

As a result of assignment diversity in the project-based class, the evaluator spent considerable time inspecting each assignment to determine the level of competence for each learning objective. Making this even more difficult was the fact that the same learning objectives in the project-based group were covered by multiple assignments from the traditionally taught class spanning a different period.

D. Student Frustration

Another challenging aspect of the project-based homework was that students would often get frustrated in the beginning because they were unsure how to achieve requirements that had not yet been covered in class. As a result, some students had to rework their solutions to fit within the project requirements; we believe this turned out to be a powerful learning experience and anecdotally, students indicated in their post course evaluations they preferred this type of learning because it felt more like the real world.

V. FUTURE WORK

We are encouraged by the positive results of this initial research experiment and believe that using a project-based approach, even in very introductory CS classes, can lead to better engagement and understanding when compared to more traditional methods. We would like to take this work further and see how this methodology applies to different CS courses and a larger scale.

ACKNOWLEDGMENT

The authors would like to thank the instructor of the control group, Patricia Duce, for sharing her materials with us. We also thank the graduate grader, James Stauder, for all his hard work assessing student work.

REFERENCES

[1] Reges, S. (2006). Back to Basics in CS1 and CS2 *SIGCSE 2006*.

[2] Rader, C., Hakkarinen, D., Moskal, B., Hellman, K. (2011). Exploring the Appeal of Socially Relevant Computing: Are Students Interested in Socially Relevant Problems? *SIGCSE 2011*

[3] Holladay, S. (2016). Projects lead to principles: applying project-based learning to animation. *November 2016 SA '16: SIGGRAPH ASIA 2016 Symposium on Education: Talks*

[4] Layman, L., Williams, L., Slaten, K. (2007). Note to Self: Make Assignments Meaningful. *SIGCSE 2007*

[5] Sung, K., Rosenberg,R., Panitz, M., Anderson, R. (2008). Assessing game-themed programming assignments for CS1/2 courses. *GDCSE '08 Proceedings of the 3rd international conference on Game development in computer science education*, 51-55.

[6] Slivovsky, L., Liddicoat, A., Clark, C., Widmann, J., Mello, J., Self, B. (2007). Enabling creativity in capstone design. *SoD '07: Proceedings of the 2007 Symposium on Science of Design*.

[7] Barnes, T., Chaffin, A., Godwin, A., Richter, H. Game2Learn: Building CS1 learning games for retention. *ITISCE 07*.

[8] Forte, A., Guzdial, M. (2005). Motivation and Non-Majors in Computer Science: Identifying Discrete Audiences for Introductory Courses. *IEEE Transactions on Education*. *48(2)*, 248-253.

[9] Guzdial, M. (2003). Computers for Communication, Not Calculation: Media as a Motivation and Context for Learning. *Proceedings of 37th Hawaiian International Conference of Systems Sciences. ITiCSE Proceedings*, 104-108.

[10] Vega, C., Jiménez, C., Villalobos, J.. (2013). A scalable and incremental project-based learning approach for CS1/CS2 courses. *Education and Information Technologies. 18*. 309-329. 10.1007/s10639-012-9242-8.

[11] Beres, P. J., (2011). Project-Based Learning and its Effect on Motivation In the Adolescent Mathematics Classroom, *Education and Human Development Master's Theses*. 39.

[12] Helle, L. (2006). Project-Based Learning in Post-Secondary Education – Theory, Practice and Rubber Sling Shots. *Higher Education*, Volume 51, Issue 2, 287-314

[13] Larmer, J., Mergendoller, J., Boss S. (2015). *Setting the Standard for Project Based Learning: A Proven Approach to Rigorous Classroom Instruction.*

[14] Lee, J. S., Blackwell, S., Drake, J., Moran, K. A. (2014). Taking a Leap of Faith: Redefining Teaching and Learning in Higher Education Through Project-Based Learning. *Interdisciplinary Journal of Problem-Based Learning*, 8(2).

[15] Walker, A. (2015). *Essential Readings In Problem-based Learning*. West Lafayette, Indiana: Purdue University Press.

[16] David, L. (2008). What Research Says About Project-Based Learning, *Teaching Students to Think* Volume 65, Number 5, 80-82.

[17] Blumenfeld, P., Soloway, E., Marx, R., Krajcik, J., Guzdial, M., & Palincsar, A. (1991). Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational Psychologist*, 26(3, 4), 369–398.

[18] Lasauskiene, J., Rauduvaite, A. (2015). Project-Based Learning at University: Teaching Experiences of Lecturers, *7th World Conference on Educational Sciences, (WCES-2015)*, 788 – 792.

[19] Lewis, J., Loftus, W. (2015). *Java Software Solutions*, Ed. 8, Pearson.