2016

# Harnessing the Expended Labor of Active Learning Exercises

Amanda L. Johnson
*Middle Georgia State University*, amanda.johnson5@mga.edu

Shannon Beasley
*Middle Georgia State University*, shannon.beasley@mga.edu

# HARNESSING THE EXPENDED LABOR OF ACTIVE LEARNING EXERCISES

**Amanda L. Johnson**
Middle Georgia State University
amanda.johnson5@mga.edu

**Shannon W. S. Beasley**
Middle Georgia State University
shannon.beasley@mga.edu

## ABSTRACT

Although many faculty and students agree that engagement and learning are enhanced by using practical, hands-on exercises in the classroom, it is rare that consideration is given to the harnessing of labor expended while transacting active learning in the field of computer programming. Professors expend effort creating programming assignments that are aimed at teaching and practicing core programming concepts, and students expend effort satisfying the assignment to earn points towards a grade for the class. We propose that if secondary school teachers and educators offer up suggestions for new software needed in their classroom the resulting information can be used to create programming assignments that will produce programs satisfying the expressed need. It is in this fashion that the fruitless labor of students in programming classes can be re-purposed to create heuristic tools for use in various areas of study.

### Keywords

active learning, heuristic tools, repurposing, gamification, educational software, supply chain, experiential learning

## INTRODUCTION

There are many concepts, that have become so ingrained in everyday thought that others will complete the idiom for you when you announce the thought process. If you proclaim out loud that “one should not work harder,” someone will quickly finish the observation by announcing that you should “work smarter.” While many educators and students acknowledge that learning is best supported and practiced through doing, many of these same educators fail to harness the effort expended by students when completing the active learning exercises. For example, a culinary student learns their potential profession from cooking, but the end result is not simply disposed of without being eaten by someone to avoid waste. On the other hand, in the act of transacting computer programming education, many exercises yield fruitless results beyond satisfying a learning objective. These hands-on exercises can be viewed as an opportunity for educators to work smarter and harness an unused source of labor for creating educational software.

In terms of cost benefit analysis or return on investment, the potential benefits to be realized are tremendous. Although, this observation seems to be akin to endorsing slave or conscripted labor, it must be noted that all parties are benefiting from the endeavor; and the labor is already in progress only to be disposed once the quality of the programming assignment is assessed. If professors teaching computer programming classes are willing to collaborate with colleagues in secondary education and motivated programming students, this source of “free” labor can be harnessed or re-purposed to satisfy new goals. This flow of information in all directions forms the basis of a supply chain for creating re-usable software that grows with the student and prospective teacher. Since this model encourages an open flow of productive observations, a feedback mechanism is created for all parties.

Many Information Technology (IT) students have indicated in previous studies that they prefer a blend of hands-on learning combined with traditional pedagogical approaches (Floyd, Harrington, and Santiago, 1). The advantages to this active or experiential learning approach include increased student engagement, improved problem solving and critical thinking skills, and increased opportunities for satisfying multiple learning styles (Kinsley, 6; Kinsley 7). Active learning involves going beyond mentally considering a concept presented in the classroom by actually applying the concept to observe a resulting outcome (Gerard, Knott, and Lederman, 2). The student can then go beyond learning through doing by reflecting on the active learning experience or real world experience to achieve experiential learning outcomes (Kolb, 8). In the case of creating educational programs, it is often beneficial to add the concepts and practices often found in games to produce a gamified approach that increases engagement. Gamified environments create a comfortable learning tool for students who are accustomed to being immersed in game settings (Gunter, Kenny, and Vick, 3; Kapp, 5).

In the case of teaching IT students the art of computer programming, several tasks are necessary to guide the student from the pseudocode or problem description through program design and coding a finished program (Zhao, 15). This task becomes much easier if the IT student believes that the work being done is more than just an academic exercise. This can be accomplished by providing the student with the knowledge that the exercises produce software that will be used in the secondary school classroom, and the students' resume is changing from passive skills to active statements describing the software the IT student has created. The end result is increased IT student engagement and an increase in self-confidence and sense of accomplishment for the IT student (Smart, Witt, and Scott, 13).

In general, a supply chain is a series of steps that lead from raw materials through a series of manufacturing processes to reach a finished product. The synergistic relationship that develops between elements within the supply chain allows for the realization of benefits created by the exchange of information and sharing of resources (Melnyk, Narisimham, DeCampos, 9). It is this form of outsourcing work elements that lead us to propose that active learning by programming students constitutes a source of fruitless labor that can be repurposed to create a fruitful result. We categorize labor expended as an academic endeavor with no other useful benefit as fruitless because the product will cease to have value once a grade is generated and recorded. However, if the fruit of an exercise is a program that can be used by other students at different levels or in different fields, the result is now a useful fruit or program.

## CREATE A SUPPLY CHAIN TO HARNESS FRUITLESS PROGRAMMING LABOR

For the purposes of this discussion, the terminology used to describe various players will be determined by their role in the cycle rather than their level of education or employment position. It is our assertion that all potential stakeholders will be viewed as potential learners in some fashion during participation in the supply chain. This assertion is due to the fact that communication exists directly or by proxy between all stakeholders. Lecturers, instructors, and professors that teach college programming classes will be referred to collectively as college professors. Students studying information technology in college classes will be referred to as IT students. Secondary school teachers will be identified as teachers and their students will simply be called students. Collectively the term educators will encompass the group of teachers, professors, and any applicable support staff.

To realize this potential benefit, the following steps are suggested (See Figure 1): 1) A college professor receives a suggestion for needed software from a secondary instructor. 2) IT students are assigned exercises that produce production software to attain programming skills. 3) Secondary instructors receive software that satisfies a need within the classroom. 4) The educational software is then utilized by the secondary instructor to provide learning to the secondary students. Each stage of the cycle feeds into the next transforming the once fruitless labor into a rewarding and beneficial experience for all participants. The cycle also provides a feedback loop that extends to all participants, providing opportunities for continuous improvement and new opportunities as the cycle starts again.

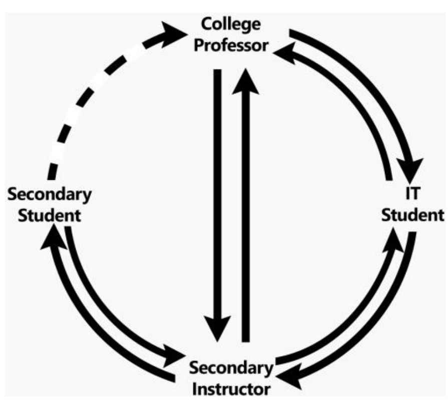


Figure 1. Information Flow

It is important to note when looking at the flow of information in figure 1, that a certain degree of compartmentalization does exist to promote the autonomy of individuals while enhancing information exchange. In other words, the secondary instructor is treated as a subject matter expert (SME) within their sphere of influence. The college professor is not telling the secondary instructor how or what to teach. The college professor is asking how existing teaching elements can be transformed from paper or perishable elements into digital components that will evolve and can be re-used with subsequent class offerings.

The effectiveness and value realized by utilizing a supply chain approach lie in the relevance, accuracy, and timeliness of information exchanged with partners in the supply chain (Melnyk, Narisimham, DeCampos, 9). For example, if a child doesn't like a software tool created by an IT student, but never tells the secondary instructor why the software is disliked, the secondary instructor is limited in the constructive feedback that can be provided to the college professor. In a similar fashion this limitation is present to potentially diminish productivity in the supply chain if effective communication is not encouraged by all parties. This also serves to foster a relationship that has not implied or coercive power or control – everyone is equal and benefits accordingly.

It is easiest to consider the proposed cycle or supply chain in terms of the functions enacted by each character in Figure 1. The college professor engaged in teaching computer programming classes is a likely candidate for creating the supply chain (starting the proposed project). The college professor meets with a secondary instructor and inquires if the secondary instructor has any teaching tools that have been: a) tested over time b) determined to be useful as an educational tool c) that can be transformed from a physical tool into the digital tool or game. This information is valuable to the college professor because it becomes the basis for creating new active learning assignments supporting the teaching of computer programming. Additionally, the secondary teacher and students become testers of the new software and create a feedback loop for continuous improvement. The secondary teacher benefits from eliminating the use of perishable items as teaching tools and the associated labor involved in preparing/creating these tools. The secondary student benefits from increased availability and the re-usable nature of digital tools. Also, the feedback loop and continued improvement allows the tool/game to evolve along with the secondary student in the form of user interface similarity as the secondary student progresses through subsequent grade levels. The IT student has the satisfaction of knowing that time and labor expended amounts to more than merely earning a grade. Harnessing the full potential of the labor expended from IT students requires a certain amount of buy-in by all involved, but the project does not require participation of every potential stakeholder to be successful.

A pilot study was performed with a small group of four IT students from a southeastern university in conjunction with a professor teaching computer programming to determine the feasibility of the proposed model and supply chain. After a team of willing participants was assembled for the pilot study, an elementary school teacher from the same area that had previously expressed a need for a new educational program was contacted to establish a meeting. During the meeting the secondary teacher provided the pilot team with a flip book used to teach various competencies to kindergarten students. The only instructions found in the book involved a request that parents select one activity per day to perform with their child for ten minutes. The objectives found in the flip book included counting, addition, spelling, shape recognition, basic civics, and other things. During the weeks following the meeting, the four students were then tasked with brainstorming to determine ways to convert various activities in the flip book into the gamified learning software. The college professor asked the pilot students to consider the following while brainstorming: 1) The software must be engaged for learners and not simply desirable as a game. 2) The software should have the potential to increase in complexity as the students' level of knowledge increases. 3) The software should avoid using sound or color alone as a means of interaction for conveying information. All steps to ensure accommodation of disabilities should be taken. 4) The time required to master the game controls should not exceed the time required for learning the desired content. 5) Thought should be taken to remove any elements that could be potentially offensive. 6) Feedback should be informative, but always constructive.

During the pilot study, the group of IT students was able to develop several applications using the Visual Basic programming language. Within a period of six weeks, the IT students used the concepts taught throughout the semester to create an application as an initial educational tool to teach and allow students to practice counting, addition, and spelling. Through the direction of the college professor, the IT students were able to expatiate upon the basic tools to include an introductory navigation page. This page was projected to address the ability of a secondary instructor to control the order in which the games could be accessed, For example, a hopscotch diagram was the initial design. With each block of the hopscotch, the secondary student must successfully complete a predetermined percentage of problems before being able to progress to the next hopscotch square. Another IT student from the pilot group used skills that were being learned through class to advance a simple dice, counting game to progress from two dice up to four dice. This game allowed the secondary student to learn to count and add beyond the restrictions of the number twelve. The same game was then advanced by requiring the student to spell out the answer. This initial pilot study was sufficient to convince the students and the professor that this project has practical benefits and can be considered useful for all stakeholders involved.

Following the success of the pilot study, the college professor and one of the original students will create a new team with three new computer programming students to implement the proposed supply chain for creating educational software in conjunction with a local elementary school. The process will begin by meeting with representatives from the elementary school at the end of the academic year to collect ideas for potential software that may be needed. The IT students will create, test, and debug software based on the information provided over the summer break. The secondary instructor will be provided a working prototype prior to the beginning of the school year with enough available time for testing. During the following year, the feedback loop and relationships will be refined to produce a functioning supply chain. Beginning with the second year of the project, consideration can be given to evaluating the effectiveness of the software in supporting active learning.

### CONCLUSION

Once you realize that computer programming assignments represent an untapped source of potential labor for creating educational tools and other software, it simply becomes a matter of determining the organization of a system that will produce the maximum potential benefit for all stakeholders. It is somewhat appropriate that an active learning activity becomes the tool that creates an active learning tool. As we noted earlier, this project does not require the participation of all potential stakeholders, but merely a subset of stakeholders is sufficient to realize a benefit. However, buy-in on a large scale allows programmers to create educational interfaces and games that are re-usable across multiple grade levels. This prevents the need for a student to have to learn a new interface or game every time new information is presented.

## REFERENCES

1. Floyd, K. S., Harrington, S. J., and Santiago, J. (2009). Improving I.S. student engagement and perceived course value. Proceedings of the 2009 Southern Association for Information Systems Conference, 24-29.
2. Gerard, PhD, J., Knott PhD, M., and Lederman, M.Ed, R. (2012). Three Examples Using Tablet Technology in an Active Learning Classroom: Strategies for Active Learning Course Design Using Tablet Technology. *Global Education Journal*, *2012*(4), 91-114
3. Gunter, G., Kenny, R., and Vick, E. (2007). Taking educational games seriously: Using the RETAIN model to design endogenous fantasy into standalone educational games. *Education Tech Research Dev Educational Technology Research and Development*, *56*, 511-537.
4. Ismail, M., Ngah, N., and Umar, I. (2010). Instructional Strategy in the Teaching of Computer Programming: A Need Assessment Analysis. *The Turkish Online Journal of Educational Technology, 9*(2), 125-131.
5. Kapp, K., M. (2012). The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education. Pfeiffer. ISBN 978-1-118-09634-5.
6. Kinsley, C.W. (1993). Community service learning as a Pedagogy, Equity and Excellence in Education, 26(2), 53-63.
7. Kinsley, C.W. (1994). What is community service learning? Vital Speeches of the Day. 61(2), 40-43. Kolb, A. Y. and
8. Kolb, D. A. (2005). Learning styles and learning spaces: Enhancing experiential learning in higher education. *Academy of Management Learning and Education, 4(2),* 193-212.
9. Melnyk, S., A., Narasinham, R., and DeCampos, H., A. (2013). Supply chain design: issues, challenges, frameworks, and solutions. *International Journal of Production Research, 52(7),* 1887-1896.
10. Morgan, C., and Parr, B. (2009). Using Technology to Increase Collaboration between Career Tech and Core Subject Teachers. *Technigues (Association for Career and Technical Education), 84* (8), 48-52.
11. Nicholls, D., De Charon, L., and Hutkin, R. (2010). Development of Critical Thinking and Creativity: Practical Guidelines for the Postsecondary Classroom. *American Technical Education Association Journal, 38*(1), 12-15.
12. Repenning, A. (2012). Programming goes back to school. *Communications of the ACM Commun. ACM, 55*(5), 38-40.
13. Smart, K., Witt, C., and Scott, J. (2012) Toward Learner-Centered Teaching: An Inductive Approach, *Business Communication Quarterly*, 393-394.
14. Zapatero, E. G., Maheshwari, S. K., and Chen, J. (2012). EFFECTIVENESS OF ACTIVE LEARNING ENVIRONMENT: SHOULD TESTING METHODS BE MODIFIED?. *Academy Of Educational Leadership Journal*, 16(4), 101-114.
15. Zhao, G. (2013). On effective pedagogical practice in teaching computer programming to CS/CIS majors. *Northeast Annual Decision Sciences Institute Annual Meetings Proceedings, 546 – 556.*