

An Online Multiplayer Game for Collaborative Problem Solving

Eshwar Bachu

University of the West Indies
Eshwar.Bachu@sta.uwi.edu

Margaret Bernard

University of the West Indies
Margaret.Bernard@sta.uwi.edu

ABSTRACT

Teaching computer programming to novices has always posed a major challenge to educators, most novices acquire basic programming skills but they are unable to utilize them in a meaningful way to solve many of the non-routine problems which are presented to them. As a result, they do not achieve any level of programming fluency. Collaboration has been identified as a useful tool to help overcome this problem if utilized at the right times during the programming/problem solving process. This paper presents a review of computer supported collaborative learning (CSCL) concepts, collaboration in programming and the problem solving domain. Finally, a design for a gaming application of CSCL (COPS) to help improve the problem solving ability of novice programmers is presented.

Author Keywords

Collaboration, cscl, game, multiplayer, problem solving, novice, programming, motivation, metacognition, flowchart, collaborative learning, group

INTRODUCTION

Learning to program is a difficult process for many students; this is reflected by the high failure rates in introductory programming courses around the world. This is especially worrying since programming lies at the core of information technology and increasing attention has been given to producing students who are considered fluent in information technology, most recently highlighted by the work of the author in (Sardone 2011). The authors in (Gomes and Mendes 2007) and (Deek, McHugh, and Turoff 2003) identifies low problem solving skill as one of the main reasons for this high failure rate. The results presented in (Tan, Ting, and Ling 2009) also highlights designing a program to solve a certain task as a major difficulty faced by novice programmers. Computer Supported Collaborative Learning (CSCL) has been identified as one of the most promising innovations to improve teaching and learning with the help of modern and communication technology (Lehtinen et al. 2001) since it aims to enhance learning by combining computer support and

collaborative learning (Stahl, Koschmann, and Suthers 2006). This paper proposes the design of an online multiplayer game, COPS (Collaborative Online Problem Solving), which utilizes CSCL principles to assist students with their problem solving and program design by helping strengthen their problem solving ability. Most of the successful CSCL systems have employed the use of existing technologies such as forums, whiteboards, learning managements systems and shared workspaces asynchronously but COPS encourages synchronous collaboration between learners.

BACKGROUND REVIEW

Collaborative Learning and Programming

Collaborative learning is an instruction method in which students work in groups toward a common academic goal (Gokhale 1995). Some of the major achievements of collaborative learning as described by (Slavin 1996) include (1) Motivation since the students are driven by a reward or goal and they recognize that they can only achieve their goals if the other members of the group succeed. (2) Social Cohesion implies that the students may actually care about the other members of their group. (3) Development since each member will be exposed to the different abilities and viewpoints of the individuals in the group and (4) Cognitive Elaboration since each member of the group will be required to explain their solutions in a social context and they benefit from having to provide their explanations.

In (Johnson and Johnson 1994), the following criteria is presented for tasks which are deemed applicable to collaborative learning:

- The task is complex or conceptual
- Problem Solving is desired
- Divergent thinking or creativity is desired
- Mastery or retention is important
- Quality of performance is expected
- Higher level reasoning strategies and critical thinking are needed

All of the above criteria clearly apply to computer programming and real world applications of programming are indeed collaborative. In ([Davidson 1994](#)), the following five critical attributes for successful collaborative learning are given;

1. Common Task or learning activity
2. Small group learning
3. Co-operative behavior
4. Positive Interdependence
5. Individual Accountability and responsibility

The author in ([Preston 2005](#)) used the above framework to analyze the pair programming pedagogy and concludes that pair programming is a model for collaborative learning. Pair programming research has had many successes including higher quality programs being written ([Cliburn 2003](#); [DeClue 2003](#); [Murphy et al. 2002](#)), decreased time to complete programs ([DeClue 2003](#); [Williams and Upchurch 2001](#)) and improved performance on exams([Nagappan et al. 2003](#)).

An experiment conducted by the authors in ([Bagley and Chou 2007](#)) also concluded that collaboration is an important pedagogy to use in teaching computer science and in performing java programming, their experiment investigated the influence of collaboration through pair programming and groups of three or four students. The collaboration was deemed to be most important in the 'brainstorming and formulating the problem' step of the programming process. This result follows with research that the major cause of students' failure in introductory programming is the lack of basic problem solving skill ([Gomes and Mendes 2007](#)). ([Mayer 1998](#)) describes three basic attributes which are required by a successful problem solver; basic skill / cognition which can be thought of as individual learning objectives; metaskill / metacognition which refers to knowing when and how to use the basic skills and motivation.

Computer Supported Collaborative Learning

Computer supported collaborative learning (CSCL) aims to improve the collaborative learning experience by utilizing the rapidly evolving technology available to students in the classroom. Originally, collaborative learning was mainly adopted in classroom based environments which required face to face interaction between students and lecturers but web based implementations of CSCL eliminates the need for this physical interaction. Attempts have even been made to utilize mobile phones in CSCL ([Zurita and Nussbaum 2004](#)) and to develop blended learning approaches to CSCL ([So and Bonk 2010](#)).

Although there has been an abundance of research in collaboration and programming in the pair programming pedagogy, very little research exists with CSCL and programming with the exception of the TurtleGraph system ([Jehng and Chan 1998](#)). However, a study involving the KnowCat system ([Pifarre and Cobos 2010](#)) concluded that some students' metacognitive skills increased after using the CSCL environment and that use of CSCL systems can enhance the development of metacognitive learning processes. Findings from ([Tan, Ting, and Ling 2009](#)) show that novice programmers prefer practical programming sessions and collaborating with lecturers or other students and this indicates that a CSCL application of programming can be very successful.

Using Games to Teach Programming

In ([Nickel and Barnes 2010](#)), the authors argue that traditional teaching of computer science education are not well suited to millennial students and they suggest that collaborative educational games can make the experiences better for both students and educators. A work in progress attempt at a CSCL video game is given in ([Zea, Sanchez, and Gutierrez 2009](#)) and the authors in ([Voulgari and Komis 2008](#)) agree that the principles of CSCL and problem solving can be applied to multi-player games. Most recently, ([Paraskeva, Mysirlaki, and Papagianni 2010](#)) agrees that online multiplayer games are an attractive and useful avenue for developing educational games. The use of games to teach introductory computer science and computer programming is well documented in ([Barnes et al. 2008](#); [Barnes et al. 2007](#); [Leutenegger and Edgington 2007](#); [Ceder and Yergler 2003](#); [Feldgen and Clua 2004](#); [Kahn 1999](#)). A game environment is described as one in which the concepts that emerge from interacting with it are created by the goal([Doherty and Kumar 2009](#)), the authors recognize the highly abstract nature of core programming concepts and suggests that games which are successful at teaching programming are those which causes the learner to develop and understand concepts from the content of the game as a consequence of its system and interface. Traditional learning approaches using computer support have always utilized user or learner centered design since the learner is identified as the main component but since we are dealing with a group of students and not an individual, ([Gifford and Enyedy 1999](#)) recommends that we utilize task or activity centered design principles and this is easily achieved by the use of a game which is designed around completing the objectives of the game.

GAME DESIGN

Overview

In (Deek, McHugh, and Turoff 2003), the authors present a six step model which describes the computer programming / problem solving process; formulating the problem, planning the solution, designing the solution, translation, testing and delivery. The first three steps are those which present the toughest task for novices since it requires the problem solving ability which they lack. Two common tools which are used during these steps are pseudocode and flowcharts. Pseudocode is a notation for programming which uses a combination of semi-structured programming structures and verbal instructions. Flowcharts are a visual representation of program flow using a combination of arrows and symbols (see figure 1) to represent the actions and sequence of the program. An experiment conducted in (Scanlan 1989) overwhelmingly indicated that student preferred flowcharts to pseudocode for understanding algorithms.

The web based multiplayer game COPS will allow students to collaboratively solve flowchart puzzle based problems. Flowcharts were also chosen because they depict the sequence of the program as well as the actions, these actions are the basic skills / cognition which programmers require and the sequence of the flowchart represents the metaskill / metacognition which show how the basic skills are used in a meaningful way to solve a problem. However, the use pseudocode is very important in helping students translate their solution into program code and as such pseudocode was utilized throughout the game to provide a guide for the players while solving the problem.

The target users of COPS are secondary school students and as such, a web based implementation was chosen because of the increasing familiarity of adolescents with online multiplayer games such as World of Warcraft. This implementation would also allow the students to collaborate outside of the classroom where they maybe more comfortable and express themselves more freely. Like other online multiplayer games, COPS forces synchronous communication and this will encourage higher communication activity between players (Serçe et al. 2011) which is essential for knowledge construction in CSCL.

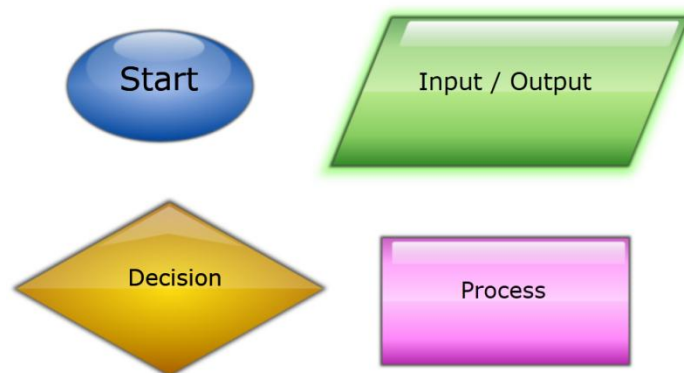


Figure 1. Flowchart Symbols

Game Play

COPS requires a group of between two to four players/students. The group will be required to create a flowchart to solve a given problem within a specified time. The game will have three levels; beginner, intermediate and expert.

At the beginner level, the students will be given all the pieces of the flowchart (symbol and text within symbol) required to solve the problem and they simply have to build the flowchart like a jigsaw puzzle placing the pieces in the correct order.

At the intermediate level, the students will be given the flowchart in incorrect order like a scrambled picture and they will be required to re-arrange the pieces to form the correct flowchart.

At the expert level, the student will be given the flowchart symbols and the text for the symbols separately and the students will be required to match the text to the correct symbol and build the flowchart. Additionally, the students will be given extra symbols and text which do not form part of the solution.

CSCL Attributes of COPS

Collaborative Learning

It is important that the critical attributes of successful collaborative learning are satisfied and the following attributes of COPS demonstrates that this was achieved.

- Each group contains between two and four students.
- All members of the group will be working on the same puzzle simultaneously. The game has a chat facility to help facilitate communication.
- Puzzles have traditionally been attempted collaboratively.

- The game will be designed as a turn based game to ensure that each student participates in the problem process, therefore, each member knows that in order to solve the puzzle and win the game, every member of the team must succeed as well. This forces the group to care about the decisions and learning process of each other.
- The design of the experiment in which the game will be tested will contain individual tests for the students to ensure that each student is accountable for themselves.

Participation and Argument

In (Weinberger and Fischer 2006), the importance of equal participation by all members of the group and argumentative discussion in the collaborative learning process is highlighted. The lack of dialogue between the members of the group in CSCL is documented in (Guzdial and Carroll 2002) and we expect to see these problems since it is prevalent in group work where certain members of the group take full responsibility and other members do not participate. These problems were addressed in the game through multiple features. Firstly, the turn based design of COPS ensures that each member of the group participates in the problem solving process, when a user makes a move; the other members of the group are polled by the game asking if everyone agrees. If the poll receives a positive result, the move is allowed else the game rejects the move and the game moves to the next player. To avoid cases of split votes, the player who made the move automatically receives a higher weighted positive vote than the other players as can be seen in table one. Additionally, when it is a player's turn, they can choose to undo previous moves before making their own move.

	Positive Vote	Negative Vote
Player making Move	1.5	N/A
Other Players	1	-1

Table 1. Weighting of Player Votes

This component of COPS which polls the members of the group will not only encourage greater participation by all group members but it will also encourage constructive argument between the players using the game's chat facility and the undoing of previous players' moves. Each player must convince their group members of their decision if the entire team is to succeed. In (Weinberger and Fischer 2006), the authors recognize that each member of the group's participation can be measured by counting

the number of chat entries made but the epistemic value of the member's contribution cannot be easily obtained from the chat logs. The epistemic dimension is not concerned with the quantity of participation but the content and value of the member's discussion related to solving the puzzle. The result of the player polls and the tracking of undoing of previous player's moves (both correct and incorrect undoing) provide a better opportunity to measure the epistemic contribution.

Coordination and Guidance

In (Wang 2009), the author recognizes the importance of proper coordination in CSCL environments to ensure that the individual efforts of the group members contribute to the learning task. This coordination is usually provided by the technology itself acting as a mediator by providing chat facilities and shared workspaces but COPS takes a more interactive approach. It is expected that the entire group may get stuck at a certain step and this can be dangerous to learning process, to help prevent this scenario, pseudocode matching the flowchart solution being constructed will be automatically generated and the students will be able to view the pseudocode version of their solution and recognize where they have miss-stepped in their solution. Another aspect of the game which will guide the learner's process is an accuracy indicator for the puzzle solution, when the group has placed a portion of the puzzle in the correct sequence, the portion will change color and players will not be able to alter any of the pieces within that portion.

Motivation

The authors in (Law, Lee, and Yu 2010) highlights the importance of motivation in computer programming courses due the uniquely demanding requirements of learning to program and motivation has also been identified as crucial to a problem solver. The use of games to provide motivation in learning and programming is well documented in (Barnes et al. 2008; Feldgen and Clua 2004; Garris, Ahlers, and Driskell 2002) and therefore the game based design of COPS will provide motivation to students. In COPS, the players are only rewarded when the entire group succeeds and all members of the group are rewarded equally. This will motivate the individual members to do their best and also encourage and help the other members of the group to do their best. Additionally in COPS, the group will be provided with a target number of moves in which to complete the puzzle and if they do, the entire group will receive bonus points. This feature of the game is intended to motivate the players to

collaborate more with their peers to ensure that the best possible moves are made throughout the game.

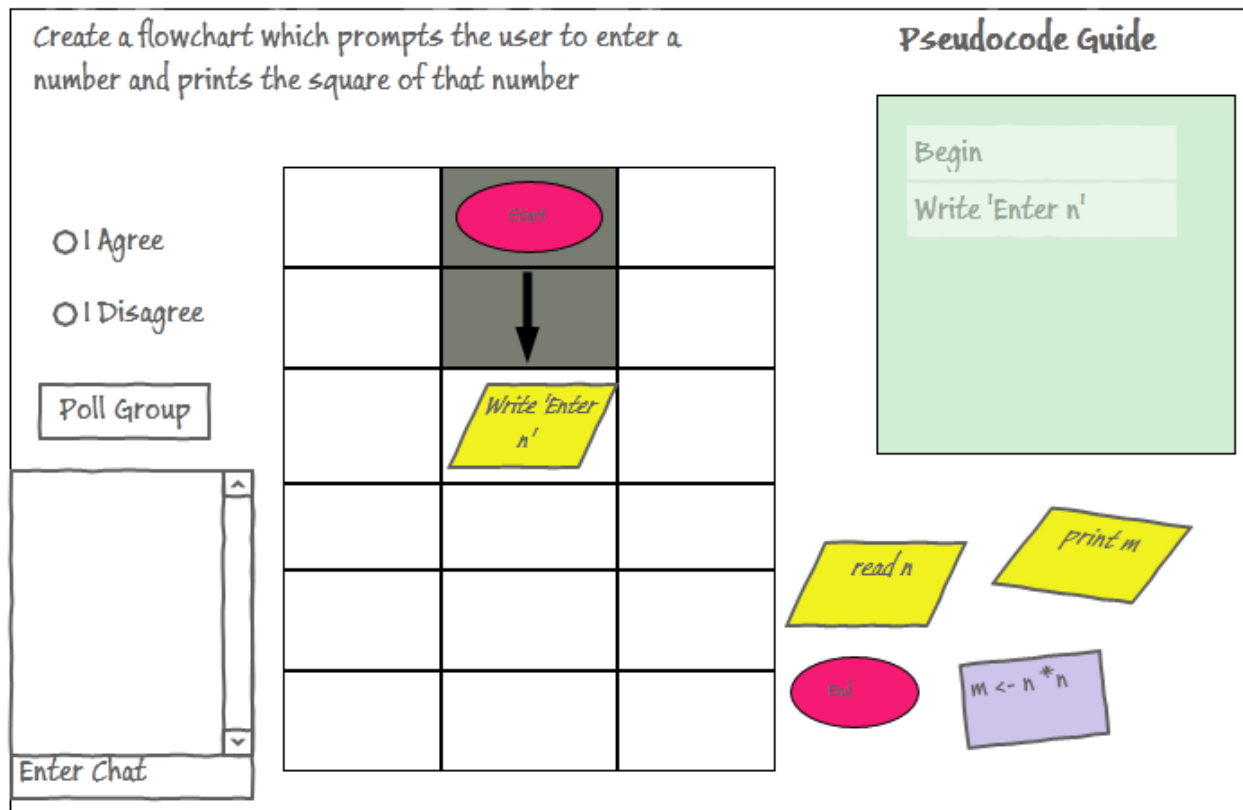


Figure 2. Screenshot of COPS Prototype.

CONCLUSION

Low problem solving skill has been identified as one of the main contributors to the difficulties faced by novice programmers. Collaboration, if utilized at the right times during the programming/problem solving process can help solve this problem as evident in research performed in pair programming. CSCL provides an avenue for enhancing this collaborative learning through the use of technology. The game COPS abides by the principles of collaborative learning and provides the cognition, metacognition and most importantly the motivation which is required by successful problem solvers in the programming domain. The various attributes of COPS are designed to force collaboration within a group in an effort to solve a problem while motivating them to ensure that the entire group succeeds since their own success lies within the success of the group. The use of games like COPS has become necessary to appeal to millennial students

and help them become better problem solvers and programmers.

REFERENCES

- Bagley, C.A., and C.C. Chou. 2007. Collaboration and the importance for novices in learning java computer programming.
- Barnes, T., H. Richter, E. Powell, A. Chaffin, and A. Godwin. 2007. Game2Learn: building CS1 learning games for retention.
- Barnes, Tiffany, Eve Powell, Amanda Chaffin, and Heather Lipford. 2008. Game2Learn: improving the motivation of CS1 students. In *Proceedings of the 3rd international conference on Game development in computer science education*. Miami, Florida: ACM.
- Ceder, V., and N. Yergler. 2003. "Teaching Programming with Python and PyGame."

- Canterbury School, Fort Wayne, IN.
Address: http://tech.canterburyschool.org/pycon/teaching_pygame.pdf.
- Cliburn, D.C. 2003. "Experiences with pair programming at a small college." *Journal of Computing Sciences in colleges* no. 19 (1):29.
- Davidson, N. . 1994. "Cooperative and collaborative learning: an integrated perspective." *Creativity and Collaborative Learning: A Practical Guide to Empowering Students and Teachers*:13-30.
- DeClue, T.H. 2003. "Pair programming and pair trading: effects on learning and motivation in a CS2 course." *Journal of Computing Sciences in colleges* no. 18 (5):49-56.
- Deek, F.P., J. McHugh, and M. Turoff. 2003. Problem Solving and Cognitive Foundations for Program Development: An Integrated Model.
- Doherty, L., and V. Kumar. 2009. Teaching programming through games.
- Feldgen, M., and O. Clua. 2004. Games as a motivation for freshman students learn programming.
- Garris, R., R. Ahlers, and J.E. Driskell. 2002. "Games, motivation, and learning: A research and practice model." *Simulation & gaming* no. 33 (4):441.
- Gifford, B.R., and N.D. Enyedy. 1999. Activity centered design: Towards a theoretical framework for CSCL.
- Gokhale, A.A. 1995. "Collaborative learning enhances critical thinking."
- Gomes, A., and A.J. Mendes. 2007. "Problem solving in programming." *The Proceedings of PPIG as a Work in Progress Report*.
- Guzdial, M., and K. Carroll. 2002. Exploring the lack of dialogue in computer-supported collaborative learning.
- Jehng, J.C.J., and T.W. Chan. 1998. "Designing computer support for collaborative visual learning in the domain of computer programming." *Computers in human behavior* no. 14 (3):429-448.
- Johnson, D.W., and R.T. Johnson. 1994. *Learning together and alone. Cooperative, competitive, and individualistic learning*: Allyn and Bacon, 160 Gould Street, Needham Heights, MA 02194.
- Kahn, K. 1999. A computer game to teach programming.
- Law, K.M.Y., V. Lee, and Y.T. Yu. 2010. "Learning motivation in e-learning facilitated computer programming courses." *Computers & Education* no. 55 (1):218-228.
- Lehtinen, E., K. Hakkarainen, L. Lipponen, M. Rahikainen, and H. Muukkonen. 2001. "Computer supported collaborative learning: A review." *CL-Net Project. Retrieved April* no. 1:2004.
- Leutenegger, S., and J. Edgington. 2007. A games first approach to teaching introductory programming.
- Mayer, R.E. 1998. "Cognitive, metacognitive, and motivational aspects of problem solving." *Instructional science* no. 26 (1):49-63.
- Murphy, L., K. Blaha, T. VanDeGrift, S. Wolfman, and C. Zander. 2002. "Active and cooperative learning techniques for the computer science classroom." *Journal of Computing Sciences in colleges* no. 18 (2):92-94.
- Nagappan, N., L. Williams, M. Ferzli, E. Wiebe, K. Yang, C. Miller, and S. Balik. 2003. Improving the CS1 experience with pair programming.
- Nickel, A., and T. Barnes. 2010. Games for CS education: computer-supported collaborative learning and multiplayer games.
- Paraskeva, F., S. Mysirlaki, and A. Papagianni. 2010. "Multiplayer online games as educational tools: Facing new challenges in learning." *Computers & Education* no. 54 (2):498-505.
- Pifarre, M., and R. Cobos. 2010. "Promoting metacognitive skills through peer scaffolding in a CSCL environment." *International Journal of Computer-Supported Collaborative Learning* no. 5 (2):237-253.
- Preston, D. 2005. "Pair programming as a model of collaborative learning: a review of the research." *Journal of Computing Sciences in colleges* no. 20 (4):39-45.
- Sardone, Nancy B. 2011. "Developing Information Technology Fluency in College Students: An Investigation of Learner Environments and Learner Characteristics." *Information technology Education* no. 10:101-122.
- Scanlan, D.A. 1989. "Structured flowcharts outperform pseudocode: An experimental comparison." *Software, IEEE* no. 6 (5):28-36.
- Serçe, F.C., K. Swigger, F.N. Alpaslan, R. Brazile, G. Dafoulas, and V. Lopez. 2011. "Online collaboration: Collaborative behavior patterns and factors affecting globally distributed team performance." *Computers in human behavior* no. 27 (1):490-503.
- Slavin, R.E. 1996. "Research on cooperative learning and achievement: What we know, what we

- need to know." *Contemporary educational psychology* no. 21:43-69.
- So, H.J., and C.J. Bonk. 2010. "Examining the Roles of Blended Learning Approaches in Computer-Supported Collaborative Learning (CSCL) Environments: A Delphi Study." *Subscription Prices and Ordering Information*:189.
- Stahl, G., T. Koschmann, and D. Suthers. 2006. "CSCL: An historical perspective." *Cambridge handbook of*.
- Tan, P.H., C.Y. Ting, and S.W. Ling. 2009. Learning Difficulties in Programming Courses: Undergraduates' Perspective and Perception.
- Voulgari, I., and V. Komis. 2008. Massively multi-user online games: The emergence of effective collaborative activities for learning. Paper read at Second IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning.
- Wang, Q. 2009. "Design and evaluation of a collaborative learning environment." *Computers & Education* no. 53 (4):1138-1146.
- Weinberger, A., and F. Fischer. 2006. "A framework to analyze argumentative knowledge construction in computer-supported collaborative learning." *Computers & Education* no. 46 (1):71-95.
- Williams, Laurie, and Richard L. Upchurch. 2001. "In support of student pair-programming." *SIGCSE Bull.* no. 33 (1):327-331. doi: 10.1145/366413.364614.
- Zea, N.P., J.L.G. Sanchez, and F.L. Gutierrez. 2009. Collaborative learning by means of video games: an entertainment system in the learning processes.
- Zurita, G., and M. Nussbaum. 2004. "Computer supported collaborative learning using wirelessly interconnected handheld computers." *Computers & Education* no. 42 (3):289-314.