121st ASEE Annual Conference & Exposition
Indianapolis, IN
June 15-18, 2014
Paper ID #8722

Using Interdisciplinary Game-based Learning to Develop Problem Solving and Writing Skills

Dr. Reneta Davina Lansiquot, New York City College of Technology

Reneta D. Lansiquot is Associate Professor of English and Assistant Director of the Honors Scholars Program where she earned her first degrees, an A.A.S. in Computer Information Systems and a B. Tech in Computer Systems, New York City College of Technology, City University of New York. She earned her Ph.D. in Educational Communication and Technology at New York University after completing her M.S. in Integrated Digital Media at Polytechnic University (now The Polytechnic School of Engineering, or NYU Engineering). Her mixed-methodology research focuses on interdisciplinary studies. She has presented her research at numerous national and international conferences in Austria, Canada, Greece, Japan, and Portugal and has published peer-reviewed book chapters and journal articles on technical writing, game design, virtual reality, and problem-solving across the disciplines. Her book is entitled Cases on Interdisciplinary Research Trends in Science, Technology, Engineering, and Mathematics: Studies on Urban Classrooms.

Dr. Ashwin Satyanarayana, New York City College of Technology

Ashwin Satyanarayana earned his M.S. and Ph.D. in Computer Science from the University at Albany, State University of New York. He has been a Research Software Developer at Microsoft, where he worked on Microsoft's search engine Bing. His research interests include data mining, machine learning, data preparation, information theory, and applied probability with applications in real-world learning problems.

Dr. Candido Cabo, New York City College of Technology/CUNY

Candido Cabo earned the degree of Ingeniero Superior de Telecomunicacion from the Universidad Politecnica de Madrid in 1982, and a Ph.D. in Biomedical Engineering from Duke University in 1992. He was a post-doctoral fellow at Upstate Medical Center, State University of New York, and a research scientist in the Department of Pharmacology at the College of Physicians and Surgeons of Columbia University. In 2000, he joined New York City College of Technology, City University of New York (CUNY) where he is a Professor in the Department of Computer Systems Technology. Since 2005, he has been a member of the doctoral faculty at the CUNY Graduate Center. His research interests include computer science and engineering education and the use of computational models to understand and solve problems in biology.

Using Interdisciplinary Game-based Learning to Develop Problem Solving and Writing Skills

Abstract

Students majoring in computing and engineering fields generally perceive that courses in their major are not related to the general education (liberal arts and sciences) courses required for their degree. This separation prevents the transfer of skills between courses in the major and general education courses that could result in mutually beneficial synergies. To challenge their preconceptions and to help students develop connections between major and general education courses, we developed a learning community that links two courses in Computer Systems (one course is an introductory course to problem solving and computer programming, CS1, and the second course is an introduction to the field of computer systems, CS0) and English Composition (EG1).

In this paper we describe an innovative approach to the teaching of computing and writing to first-year students majoring in a Computer Systems degree at a college of technology. The theme of the learning community is the development of narratives (a plot or schematic structuring of temporal actions) and their implementation as a video game prototype. Common student learning objectives and general education student learning outcomes for our courses include: use creativity to solve problems; understand and navigate systems; work productively within and across disciplines; use the tools needed for communication, inquiry, creativity, and analysis; gather, interpret, evaluate, and apply information discerningly from a variety of sources; and communication in diverse settings and groups, using writing (both reading and writing), oral (both speaking and listening), and visual means.

In the English composition class, students write original video game narratives in groups; in their CS1 computer programming class students implement these stories using *Alice*, a computer programming environment that supports the creation of three-dimensional animations; and, in the CS0 survey course, students explore architectural and hardware issues to describe a possible game delivery platform. The concepts and skills introduced in the computer courses are contextualized by a problem (game design) that is relevant to students and connected to concepts and skills developed in the writing course. Moreover, traditional English composition is taught to connect to the computing courses that first-year students take. The common student assignment across the three courses in this learning community is a game design document which includes analysis (background and problem description, target audience, review of existing projects and media selection), design (user characteristics, goals and objectives, and description of the delivery platform), and project description (narrative of project design, review of relevant literature, flowchart of the entire project, and storyboards). When given the chance to work on a meaningful project of their own choosing, students collaboratively created video game prototypes by leveraging their problem-solving, programming, and writing abilities gained in these three courses.

1. Introduction

Students majoring in computing and engineering fields at our institution must take a combination of courses within their discipline (about two-thirds of the total credits required for a degree) and general education courses (liberal arts and sciences, comprising the remaining one-third of the total credits) in order to graduate. Each degree program has well-defined learning outcomes that are mapped to the learning outcomes of each individual course within the program. Students are expected to perceive the different courses required for graduation as part of a single whole (i.e., the degree program), and also to establish connections and synergies between the different components (courses) of the curriculum.

Unfortunately, the transfer of concepts and skills between courses is lacking. Students tend to perceive courses in their major and general education courses as unrelated to each other; as a consequence, they are not able to transfer skills between those courses. This lack of transfer is not a problem unique to our institution. In fact, there is abundant evidence that transfer of skills between courses is relatively rare.[2,3,6] The problem occurs not only in the crossover between general education and technologically oriented courses, but even between different courses of the same type. The lack of transfer is likely due to multiple factors. Students may have forgotten some of the material learned in a previous course; students may not perceive the connections; students may see the connections but are unable to use the material in meaningful ways in a different context; or the pedagogical approach used by instructors may not be conducive to transfer.[3]

Approaches used to facilitate transfer of learning include the use of reflective writings, contextualization of learning experiences, and application of learning to real life. Multiple strategies have been suggested to encourage transfer [3]: making the need for transfer of learning explicit to students, advising students to take courses in the appropriate sequence, emphasizing in each course the material that students need to transfer to other courses, "practicing" transfer by inviting guest lecturers, development of metacognitive skills, and reinforcing concepts by using them often and in different contexts. Regardless of the strategies used, it seems apparent that transfer of learning does not occur automatically and that curriculum and course design should intentionally emphasize the connection between courses to stimulate transfer.

EG1, English Composition, is required of all students at the college. Likewise, CS1, Problem Solving with Computer Programming, is a required course for first-year students, and with a companion course, CS0, Introduction to Computer Systems, is a prerequisite for all other computer courses in the major. These gateway courses lay a foundation, generally, for the rest of the academic courses in college, and specially, for more advanced work in computer programming, database, and networking courses. To make more obvious to students the connections between those three courses, and to facilitate transfer of skills between those courses and beyond, we have developed a learning community (LC) that links the three courses as part of our innovative approach to the teaching of computing and writing to first-year computer systems majors at a college of technology. The theme of the LC is programming narratives—that is, having students write narratives that they will then implement, using computer programming, as a video game prototype. We believe that the LC approach incorporates and builds on many of the suggestions in the literature on how to facilitate transfer; moreover, it makes a statement, early in the students' academic careers, about the importance of connecting courses in the major and those in general education so as to facilitate transfer. This LC also builds on previous research showing that introducing narrative elements into problem-solving courses improves student performance in general as well as in computer programming-related problem-solving skills.[4,9]

We begin this paper by introducing the concept of a first-year LC, along with the learning outcomes and objectives of the three courses which are part of our LC. Next we describe the common assignment shared between the three courses, followed by the evaluation criteria for the common assignment. We then present data comparing the performance of the LC students with other students who took the same courses but were not part of the LC. The paper concludes with a discussion of the findings.

2. First-Year Learning Community (LC)

A LC is a group of students who enroll in two or more courses, generally in different disciplines that are linked together by a common theme, in an academic semester.[8] LCs are one of the ten high-impact educational practices recognized nationally to improve student persistence using data from assessment to increase retention. [8] First-year students are enrolled in two or three courses where faculty work together to provide a supportive structure through collaborative learning experiences and peer mentoring fosters better academic achievement.

Our institution is one of the most racially, ethnically, and culturally diverse institutions of higher education in the northeast United States: 31.5% of our students are African American, 33.8% are Latino, 20% are Asian or Pacific Islanders, 11.3% are Caucasian, and 0.6% are Native Americans. The College's fall 2013 enrollment was 16,861.

At the College, LC students are recruited randomly and then given a list of linked course options in which enrollment is first-come, first-served. Students enrolled in LCs at the College are provided with social and education networks to support learning. Activities include a Welcome Orientation, registration workshops, study rooms, a mid-semester social event, and participation in a peer program. Faculty members are trained to implement cooperative learning, alternative assessment in the classroom, cross-disciplinary writing assignments, and critical thinking activities. They also learn how to make use of the campus's counseling, library, and other educational resources as well as how to incorporate technology in the learning process.
We have implemented LCs at our institution for more than 10 years, and the academic performance of students participating in LCs reflects the national trends. When compared to the general population at the College, students in LCs earn higher GPAs, have higher retention rates, and demonstrate greater satisfaction[1].

With these results in mind, we developed a LC entitled "Story-Telling in Role-Playing and Action-Adventure Games." In this LC, students gain insight into how modern video games are designed (both hardware and software) using narratives. As part of a LC—which includes CS0 (Introduction to Computer Systems), CS1 (Problem Solving with Computer Programming) and EG1 (English Composition I)—students leverage the problem-solving, computer programming, and writing skills gained in these three courses to produce a videogame prototype. This prototype includes a sample game world, characters, their interactions with their setting, and a narrative establishing non-linearity.

2.1 English Composition I (EG1)

At the College, a typical EG1 course does not have a video game narrative theme and students do not have the opportunity to implement their stories; however, both courses help students to develop their ideas by using rhetorical modes including narration, comparison/contrast, analysis, and argumentation.

In this composition course, students write a series of assignments that result in a coherent exposition or narrative (i.e., they relate a set of events that follow a logical and coherent direction so as to form a story). The first third of this fifteen-week semester course provides instruction on basic elements of a story, using examples from different literary and media genres. Students focus first on the elements of drama in plays and short stories, then explore examples of the plot structure of films, and are finally introduced to the idea of the mythic hero and the hero's journey, which can be depicted as a circle. The explanation of the hero's quest begins at the top of the circle with the Ordinary World and then moves counterclockwise through a series of stages: The Call to Adventure; The Refusal of the Call; Meeting the Mentor; Crossing the Threshold; Tests, Allies, and Enemies; Approach; The Ordeal; The Reward; The Road Back; The Resurrection; and Return with the Elixir—which brings the hero back to the Ordinary World.[5,10] At this point, the connection to video game narratives is made and the importance of an engaging protagonist is emphasized.

During the middle five weeks of the semester, students use their knowledge of the hero's journey first to write (individually) an original video game background story, and then to pitch their story idea to their classmates to choose the best narratives to implement as a computer program, and they then revise the selected story as a group. Students (again working individually) develop an engaging character side quest, including a rationale for the importance of such a quest to the protagonist as well as to the target audience of the game (see Appendix). The detailed assignment description is provided below:

- *Create the background story for a video game, including the protagonist and antagonist:* In a three-page narrative, explain what your central character wants and what the expertise of this hero is. Why is this character on this important quest? Be sure to describe the setting—the world of your game—clearly. Provide sufficient descriptions so that it is clear what the player needs to do to "win" your game. Your video game background story should end where game play begins. Attach, on a separate page, a well-developed paragraph describing the game play. Be sure to address the following: Who is your target audience? Why should the player care about the protagonist? Why is your story socially relevant or engaging to your proposed target audience? You will present your video game plot outline to the class, and the best ideas will be chosen for further development. Therefore, consider a rationale for your game or how you would "sell" the game idea to your target audience—in this case, your classmates.

- *Revise the selected video game background story with your group:* Refine this story and broaden its scope. Each group member should contribute one additional page. The final version of this collaborative story should be at least two full pages longer than the original and should include screenshots from *Alice* (www.alice.org), a 3D programming environment.

- *Development of a side-quest:* In a three-page narrative, develop a character side quest based on your game world and the team of characters that you have created. How will game artifacts come into play? Explain how your character interacts with the world and what this character contributes to the team. Attach, on a separate page, a well-developed paragraph explaining why this side quest will be engaging to your audience. Also attach your concept map—that is, a diagram that maps the relationships among elements of the story.

Finally, during the last five weeks, students develop their game design document as a group.

2.2 Problem Solving with Computer Programming (CS1)

The second course of this LC is CS1, Problem Solving with Computer Programming. This course is designed to introduce the student to concepts of problem solving using constructs of logic inherent in computer programming languages, including procedural programming and object-oriented programming. The student learns the nature of problems, common solution approaches, and analysis techniques. During the first two weeks of the fifteen-week semester, the emphasis is on solving problems in a context known to the students—for example, navigation of mazes or games such as tic-tac-toe. Several computer programming constructs such as sequencing, selection structures, and repetition loops are introduced to solve various problems using pseudocode. During the next three weeks students solve problems with flowchart interpreters (*Visual Logic*). In the following eight weeks of the semester, students use *Alice*, which allows them to further develop their problem-solving skills. *Alice* is also used to introduce object-oriented programming concepts as the students create animations or interactive computer games. In the final two weeks of the semester, students are exposed to an IDE (Integrated Development Environment, such as NetBeans or Eclipse, which helps programmers to write, compile, and test computer programs) and basic Java programming. This exposure facilitates students' transition from the real objects that they have created and manipulated in *Alice* (i.e., characters in the game world) to the more abstract objects (like the objects found in a graphic user interface window, such as buttons, textboxes, and labels) that they will manipulate in subsequent and more advanced programming courses.

The LC common assignment (described further in the section entitled "The Common Assignment: A Game Design Document" below) is a crucial component enabling students to achieve and reinforce the learning outcomes for this course. As part of that assignment, students implement the background story for a video game developed in the EG1 class as a computer program in *Alice*. The assignment is organized around several milestones, including: (1) preparation of a flowchart of the story; (2) creating the setting of the video game with *Alice* objects; (3) creating the characters for the video game using *Alice* objects; (4) implementing the characters' interactions among themselves and with the setting, i.e., programming the story in *Alice* using methods and events; (5) developing individually a character side quest within the video game developed by the group; and (6) integrating the main story and the side quest. Through this assignment, students demonstrate their ability to solve problems with a computer, using constructs of logic inherent in computer programming languages, including procedural programming (sequencing, decision and repetition structures) and object-oriented programming (use of objects and their associated methods). Their work implementing the video game narrative in *Alice* applies to a different context the computer problem-solving skills that they acquired earlier in the course using pseudocode and a flowchart interpreter (*Visual Logic*), and it provides preparation for and a transition to learning a more formal programming language such as Java.

Incorporating narrative elements in a problem-solving course builds on previous findings that the introduction of narrative elements into problem-solving courses improves student performance in general and programming-related problem-solving skills specifically.[4,9] All sections of CS1 taught in our department now incorporate narrative elements in problem-solving computer courses through the use of *Alice*. The linking of the CS1 problem-solving course in an LC with EG1 further integrates narrative elements into computer problem-solving courses; this integration should result in improved and more transferable computer problem-solving skills.

2.3 Introduction to Computer Systems (CS0)

In this foundational course for Computer Systems, students engage in an overall inspection of the world of computing. As part of this course, students also learn introductory concepts related to the inner workings of the computer, such as operating systems, networks, and database systems. This overview of machine architecture, software development, data organization, ethics, computer security, and the theory of computing is presented to introduce students to the key threads that recur within other computer systems courses. Thus, one of the main goals of this course is to enable students to develop critical thinking skills by designing solutions to given problems using pseudocodes and algorithms.

During this class, as part of their game design document common assignment (see below), students conduct a review of relevant literature, develop pseudocode to guide them with the implementation in *Alice* of the video game prototptype, and describe the delivery platform for their video game. These components of the common assignment are crucial to enable students to achieve and reinforce the learning outcomes for this course. As part of the assignment, students are asked to:

- *Review of existing projects and identify target audience*: Reviewing involves identifying what is already known, in order to bring together results from different studies and provide a starting point for students to start on their common gaming project. This review of other games will help in identifying trends and patterns in others' research findings in the world of video games. As part of this assignment, students also identify the target audience for their particular game.

- *Media selection*: Media selection is the first step that students should consider after they come up with their video game narrative. As part of the CS0 course, students learn how video, images and audio are stored in a computer. Using this knowledge, they then address what multimedia resources they would need for their gaming project and what memory constraints these needs would present. They are also required to consider the resulting performance impact on the computer system.

- *Artificial intelligence (AI) in gaming*: As part of designing the game, students are required to complete a literature survey and identify how AI is used in their game. As part of the CS0 course, an introduction to AI is presented, including search strategies used in AI (e.g., semantic trees), robotics, and natural language interpretation. Students are then required to address these concepts in their game design.

- *Pseudocode*: Once they have their concept map ready, students write pseudocode that describes all the concrete and sequential steps in their plot. This pseudocode may include control structures (e.g., IF/ELSE, WHILE) that would be used in the implementation of the game.

- *Delivery platform*: As part of the CS0 course, students are taught about the functions of operating systems and are presented with an overview of different operating systems. Students are required to identify the delivery platform for their game.

3. The Common Assignment: A Game Design Document

3.1 Description of the Common Assignment

The common assignment across the three courses is a game design document. The common assignment is a crucial component to enable students to achieve and reinforce the learning outcomes for the individual courses that are part of the learning community. Common student learning objectives and general education student learning outcomes related to this assignment include:

- Use creativity to solve problems
- Understand and navigate systems
- Work productively within and across disciplines
- Use the tools needed for communication, inquiry, creativity, and analysis
- Gather, interpret, evaluate, and apply information discerningly from a variety of sources
- Communicate in diverse settings and groups, applying written (both reading and writing), oral (both speaking and listening), and visual means

This game design document contains three sections: analysis, design, and project description. These three sections in turn encompass the following subsections; the course in which students will complete each subsection is noted in parentheses.

- *Analysis*: background and problem description (EG1); target audience, review of existing projects, and media selection (CS0).
  In this section students first introduce their game, providing a summary of the background story and side quests and identifying their ideal player. Students compare and contrast their game to similar games. Finally, they select a delivery platform (e.g., a specific game console or computer). The problem description is added later, after students complete a literature review on artificial intelligence and human-computer interaction (HCI) and formulate a thesis related to this game design project.

- *Design*: User characteristics, content analysis, goals and objectives (EG1); description of the delivery platform (CS0).
  This section of the document notes the player characteristics; students also learn characteristics of their chosen genre (e.g., action-adventure or role-playing games). Game design elements related to these player characteristics are highlighted. The goal and objectives of the game are also stated; for instance, a student may express an intention “to

offer players a chance to immerse themselves in a fantasy RPG [role-playing video game] that has a compelling plot while experiencing the classical game-play system that has endeared itself to the hearts of thousands of gamers in their childhood." The hardware items needed to properly run the proposed game, such as specific memory requirements, graphics, and sound cards, are also presented here.

- *Project Description:* Narrative of project design, flowchart (CS1); review of relevant literature, pseudocode (CS0); concept map and storyboards (EG1).
  A step-by-step plan of how students plan to implement their game in *Alice* is the main component of this section. This includes their flowchart, pseudocode, concept map, key characters, and representative setting screen shorts from *Alice*. In addition, each member of the group provides a summary of an article on AI or HCI, and all articles are synthesized to form a relevant literature review.

3.2 Assessment of the Common Assignment

A literature review assessment rubric is used to evaluate students' understanding of current developments in the field of gaming, including their grasp of relevant research on AI and HCI. This rubric describes multiple domains and the measurement criteria for each one, including completeness of the literature review, critique of sources, synthesis of sources, writing style, and adherence to an appropriate documentation format. With regard to synthesis, the rubric measures students' ability to evaluate and report on the main ideas. They are expected not to summarize each article, but rather to organize and present the information with emphasis on its relevance to the larger topic of gaming development. Synthesizing information is not summarizing each article, but rather organizing the information by how it relates to each main point of the larger topic.

A concept map assessment rubric is used to evaluate students' understanding of the plot structure of narratives. This rubric measures important concepts and describes domains on multiple levels with regard to the breadth of net; embeddedness and interconnectedness with other concepts; and the use of descriptive links to succinctly and accurately describe all relationships. In addition, the rubric measures efficiency of the link construction in terms of how each link type is distinct from all others, clearly describing consistent relationships. Finally, it measures layout in terms of how one map is contained on a single page, whether it contains multiple clear hierarchies, how well it is laid out, and whether it provides a sufficient number of relevant examples with links.

A target audience assessment rubric is used to evaluate students' understanding of their research on identifying the target market for their game. The rubric measures how succinctly the students have defined the reasoning for choosing their specific audience (e.g., experienced game console players vs. cellular game users), characterized the size of the available market (with factual citations from magazine articles or an industry analysts), and identified other competitors who address the same needs as this product as well as any known customers for this product.

Media selection should include a thorough description of how audio, video, and images are represented in the game. Details with regard to specific file formats and software used for multimedia are expected.

A delivery platform rubric is used to determine the students' understanding of what type of gaming platform their game would need. This rubric measures how the design would consider the downloading and installation of the game on Windows, Macintosh and Linux computers.

Pseudocodes and flowcharts are evaluated on how they mimic the elements of the actual computer code, but with more focus on readability and less on technical requirements. Evaluation involves determining how the dimension variables and control structures are defined. Finally, we assess whether all the applicable tasks and elements from the narrative are contained in the pseudocode.

Students also prepare a narrative on the entire project, describing how they implemented their stories in *Alice* (implementation of the characters and setting of the story as programming objects, and their interactions using events, functions and methods). This section is evaluated based on whether the reader can reproduce what was done based on the description provided.

3.3 A Case Study

In the Appendix, we highlight assignments of one student group as a case study, providing a summary of the video game background story and side quests, a concept map of the entire game, an *Alice* screen shot showing some of the programming objects and methods used in their coding of the video game, and storyboards. All those elements were part of the group's Game Design Document.

4. Assessment of Student Performance and Satisfaction

4.1 Assessment Goals

Institutional results on LCs at our College[1] compare student success and satisfaction to the general student population, which makes it difficult to isolate the effect of our particular LC, on student success and performance in the EG1, CS1 and CS0 courses. Therefore, we had two assessment goals: to gain insight into the effect of our particular LC on students' academic performance in EG1, CS1 and CS0, and to qualitatively measure student satisfaction with the LC educational approach.

4.2 Design Methods

To quantify student performance, we compared the grades of students in the three courses of our LC ($n = 22$), with the grades of students taking the same three courses simultaneously but not as part of a LC in fall 2013 ($n = 70$). Fall 2013 was the first semester we implemented this LC with the approach described in the paper. All students taking those three courses were majors in our Computer Systems degree. For our analysis, we converted letter grades to the numeric quality points (0-4) used to calculate student GPAs. For the statistical analysis, group means differences were considered significant at the 0.05 level.

We qualitatively measured student satisfaction with the LC educational approach with a survey at the end of the fall 2013 semester.

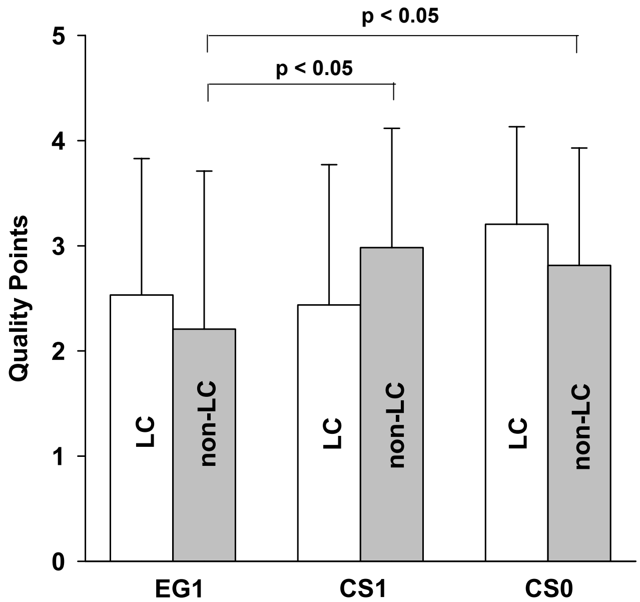

Figure 1. Means and standard deviations of numeric grades in EG1, CS1, and CS0, for students in a LC (LC in white; n=22), and not in a LC (non-LC in grey; n=70). Quality Points are the numerical conversion of letter grades used to calculate students' GPAs (range 0-4). $p < 0.05$ indicates that the group means are statistically significant.

4.3 Assessment Results

Figure 1 shows the means and standard deviations for students in the LC, and not in the LC (non-LC), for all three courses. One-way analysis of variance showed no differences between course means for students in the LC group ($p > 0.05$), but that there were differences between course means for students in the non-LC group ($p = 0.001$). Post-hoc multiple comparisons in the non-LC group showed that grades in EG1 were significantly lower than grades in CS1 ($p = 0.001$) and CS0 ($p = 0.013$). There were no differences between the grade averages for each individual course between the LC and non-LC groups.

The data in Figure 1 suggests two factors contributing to student performance: differences in performance between the computer and non-computer courses and whether students took those courses as part of a LC or not. To further study the interactions between course performance and LC, we performed two-way analysis of variance with repeated-measures. The LC factor had two levels (being part of the LC or not), and the course factor had three levels (the three grades for each student in the three courses, EG1, CS1 and CS0). The results showed that both the course factor ($p = 0.005$) and the interaction between the course and the LC ($p = 0.023$) are statistically significant. This suggests that differences in performance in the three courses are modulated by whether or not the student is taking those courses as part of a LC. These results reinforce the earlier analysis suggesting that the LC context helps students to improve their performance in EG1 to a level comparable to their performance in CS1 and CS0.

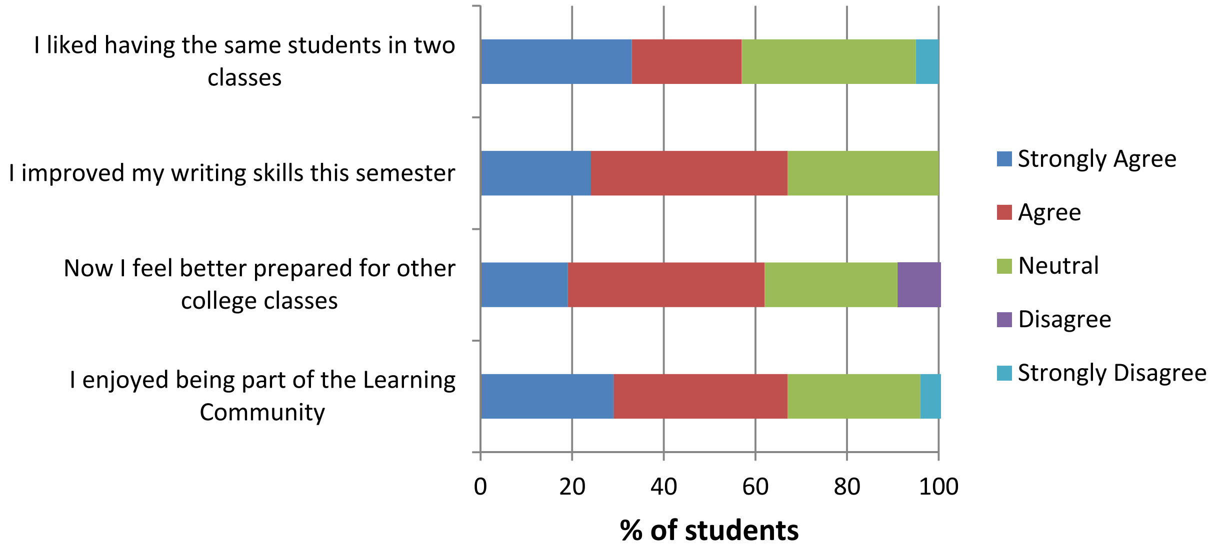

Figure 2. Results of a satisfaction survey of LC students (n=21).

Students in the LC were generally satisfied with the educational approach (Figure 2). About 67% agree or strongly agree with the statement "I enjoyed being part of the Learning Community," and about the same percentage feel better prepared for other college classes and that they improved their writing skills.

4.4 Discussion of the Assessment Results

Based on this data we can conclude that students who are not part of a LC (i.e. the majority of our students) and majoring in Computer Systems perform significantly better in the computer courses (CS1 and CS0) than in the English composition course (EG1). In contrast, for students in the LC group, performance in the EG1 course increases to the level of performance achieved in computer courses. Therefore, we propose that that increase in performance is the result of the contextualization of the learning experience that occurs in the LC: students apply writing and narrative concepts and skills to problems which are relevant to their interests and to their major. A student satisfaction survey (Figure 2) shows that students are generally satisfied with the LC educational approach. Those results are in line with institutional results on other LCs at our College.[1]

A number of factors should be considered when interpreting our results. In our analysis of student performance we have used final grades as an estimator of performance in the different courses. That is necessarily a summative indicator of performance in different types of assessments, including tests and quizzes, projects, homework assignments and student class participation. Moreover, LC courses and non-LC courses were not taught by the same instructors. Therefore, it is possible that there is some variability on grading styles by different instructors, even in courses like EG1, which have uniform finals. Further studies will be necessary to understand the effect of all those factors in our results, and to assess and compare performance in more specific concepts and skills both in writing and computing for students taking those courses in the context of a LC and outside a LC.

5. Discussion

In this paper we describe an innovative approach to the teaching of computing and writing to first-year students majoring in a Computer Systems degree at a college of technology serving mostly underrepresented minorities. We provide a framework to link introductory courses in computer systems, problem solving with computer programming, and English composition in a learning community. The theme of the learning community is the development of video game narratives and their implementation as a video game prototype using computer programs, and it provides a mechanism for concept and skill transfer between the three courses. Students in the LC develop video game narratives in the English composition course, and they implement video game prototypes in the computer courses while acquiring computer concepts and skills that will be the foundation for other courses in their major. A deliverable for all students in the LC is a game design document that is a crucial component enabling students to achieve and reinforce the learning outcomes for the three courses.

Our students tend to perceive courses in their major and general education courses as unrelated to each other and, as a consequence, they are not able to transfer skills between those courses. One of the goals of the pedagogical approach described here is to make more obvious to students the connections between the three foundational courses in the LC, contextualizing the learning experience to facilitate transfer of skills between those courses and beyond: by providing learners with meaning, concepts are embedded within a web of related concepts. Concepts with numerous connections to other concepts have greater meaning and can act as a recall cue for connected concepts. We believe that the LC approach incorporates and builds on many of the suggestions in the literature on how to facilitate transfer[3]. It also makes a statement, early in the students' academic careers, about the importance of connecting courses in the major and those in general education so as to facilitate transfer.

A second goal of our study is to investigate the effect of linking computing and writing courses on student performance in those courses. Our results show that students majoring in Computer Systems perform significantly better in introductory computer courses than in English composition courses. This may be a consequence of the students' view that computer courses and general education courses are unrelated, and that the computer courses are more "important" for their degree. However, when the English composition course is linked with introductory computer courses in a LC, students' performance in English composition improves to the same level as their performance in the computer courses. It is possible that the intentional interdisciplinary contextualization of learning that occurs in the LC helps students making connections between writing and computer courses and facilitates transfer of concepts and skills with the consequent improvement in academic performance. An earlier study by Goldfine[7] (with a different student population) showed that students with advanced skills in computer programming do not naturally make a connection between writing code and writing English, unless that connection is intentionally emphasized by the instructor, further illustrating that knowledge transfer between disciplines is rare and it must be intentionally emphasized by instructors.[2,3,6]

While the pedagogical approach described here, using LCs to link computer and writing courses, seems to be beneficial for our student population, particularly underrepresented minorities, we are not claiming that our results are general and can be easily extrapolated to other student populations and institutions. Further studies in other institutions will need to be carried out to determine how this approach might work in a different context. Moreover, it should be noted that LCs could be focused in different domains (e.g. green energy application, or integrating data from social media, etc.), which could provide students with other interdisciplinary experiences which may also pique their interest.

6. Conclusions

We can conclude: 1) Intentional interdisciplinary approaches to writing and problem solving allow students to purposefully connect and integrate knowledge and skills from across the disciplines. This integration will support students in solving problems; synthesizing and transferring knowledge across disciplines; becoming flexible, reflective thinkers who are comfortable with complexity; and thinking critically, communicating effectively, and working collaboratively, skills that will prepare our students to be lifelong learners in their careers; 2) Students majoring in Computer Systems perform better in introductory computer courses than in English composition courses; 3) Linking English composition courses with computer courses in the interdisciplinary context of a LC results in the improvement of students' performance in English composition to the same level of their performance in the computer courses.

7. References

[1] Assessment and Institutional Research. (2010). *CUNY Student Experience Survey*. New York City College of Technology, CUNY.
[2] Barnett, S. & Ceci, S (2002). When and where do we apply what we learn? A taxonomy for far transfer. *Psychological Bulletin*, *128*(4), 612-637.
[3] Benander, R., & Lightner, R. (2005). Promoting transfer of learning: Connecting general education courses. *The Journal of General Education*, *54* (3), 199-208.
[4] Cabo, C., & Lansiquot, R. D. (2013). Development of interdisciplinary problem-solving strategies through games and computer simulations. In R. D. Lansiquot (Ed.) *Cases on interdisciplinary research trends in science, technology, engineering, and mathematics: Studies on urban classrooms* (pp. 268-294). New York: IGI Global.
[5] Campbell, J. (1949). *The hero with a thousand faces*. Navato, CA: New World Library.
[6] DeCorte, E. (2003). Transfer as the productive use of acquired knowledge, skills, and motivations. *Current Directions in Psychological Science*, *12* (4), 142-146.
[7] Goldfine, R. A. (2004). Computer science majors' perception of the overlapping cognitive structures between computer programming and English composition. Ph.D. Dissertation. Georgia State University, Atlanta, GA, USA. AAI3169747.
[8] Kuh, G. D. (2008). High-impact educational practices. Washington, D.C.: AAC&U.
[9] Lansiquot, R. D., & Cabo, C. (2011). Alice's adventures in programming narratives. In C. Wankel & R. Hinrichs (Eds.), *Transforming virtual learning: Cutting-edge technologies in higher education* (Vol. 4, pp. 311-331). Bingley, UK: Emerald.
[10] Vogler, C. (2007). *The writer's journey: Mythic structures for writers* (3rd ed.). Studio City, CA: Michael Wiese.

Acknowledgements

We would like to thank Learning Communities Director, Estela Rojas, for providing general data on the program at the College as well as Coordinated Undergraduate Education (CUE) First-Year Program Director, Lauri Shemaria-Aguirre for taking time out of her busy schedule to meet with us to discuss the recruitment process and support structure for learning communities. Moreover, we would also like to thank Peer Mentors, Tamrah Cunningham, Elaine Green, and Walter Rada, all former students in our learning community, for helping students adjust to college life. In addition, we would like to extend our gratitude to the Assessment and Institutional Research (AIR) office for providing requested data. Finally, we would like to thank Hong Li, Chair of the Computer Systems Technology Department, and Nina Bannett, Chair of the English Department, for their support.

Appendix

Below is a summary of student Noel Melendez's video game background story *Schrödinger's Workshop*, which was chosen by his classmates to be further developed and implemented in *Alice*. With help from Noel's group members, Tykila McCray, Hilario Salas, and Olajide Odunaike, his original background story was revised and each member of the group wrote side quest narratives, which is also summarized below, and represented in a sample concept map (Figure A1). References to corresponding screenshots from *Alice* have been added throughout the summary for purposes of illustration.

Schrödinger's Workshop
Noel Melendez, Tykila McCray, Hilario Salas, and Olajide Odunaike

During the prologue of the video game prototype called Schrödinger's Workshop, the player learns that he or she is in control of the main protagonist, Lloyd Deteget, a teenage boy genius. The name "Lloyd" is a play on the word "alloy," or the solution of two or several metals; the last name, Deteget, means "to unveil" in Latin. By following Lloyd on a typical afternoon, the player learns that he is studying at the most prestigious engineering college in the city. He or she also learns that Lloyd has a habit of leaving the safety of his home to collect parts and metal scrap for his strange inventions. The lack of parental supervision throughout the prologue suggests that Lloyd is an orphan.

When our hero takes a brisk walk outside, the player sees that the world of Schrödinger's Workshop is a vast utopia contained in a tight metropolitan setting, littered with advanced technologies such as flying vehicles, hover cars, and brilliant lights (see Figure A2). The city is protected by a large, cascading metal wall that shields it from the mysterious, foggy environment lying beyond it. The story line explains that, if someone were to exit the city's safety and venture into the fog, he or she would be snatched by men robed in white, never to be seen again. This myth has been passed down from generation to generation, but Lloyd has always thought that it was a hoax and wants to explore the outside world.

On his way to the junk yard in search of scrap metal, Lloyd finds that his headphones are malfunctioning. Soon radio transmissions are interrupted and LCD screens displayed in window shops flicker violently. Lloyd hears a message announced by a disappointed female voice: "We are with heavy hearts today as bring this news to you. We are pulling the plug; this experiment was unsuccessful." The sun suddenly dims, all technological devices cease to function, and vehicles drop from the sky like flies. The city is now in darkness, except for fires from the quickly accumulating accidents (see Figure A3). One falling aircraft damages the city wall, creating an escape opportunity for Lloyd, who ventures out into the fog (see Figure A4).

Outside the city, the terrain seems bouncy, plastic as if manufactured, and unnaturally flat. Eventually Lloyd encounters a large, intimidating wall that stretches in all directions, with an entrance to a building called the Cypress Laboratory, which will be the main focus of the game (see Figure A5). Peering inside the laboratory, Lloyd sees men and women robed in white, scurrying around and recording data from LCD screens that litter the cold and sterile halls.

It turns out that the laboratory has been conducting an experiment on Lloyd's town, which they called Schrödinger's Workshop. The "sun" was an artificial source of light, and the city has been observed for several centuries through a series of monitors in the laboratory until the power was cut off.

The game has several side quests. Each one has an effect on the game and also reinforces the idea that the player has control over the fate of many people, not just Lloyd. For example, in "Dean's Lament," Lloyd visits his alma mater and finds that the college dean, named Hunnivant, is in physical peril. Hunnivant reveals to Lloyd that he was involved with the man responsible for conducting the grand experiment on Lloyd's town. The player must help the dean to receive medical attention and press him for information. Whether the player ignores Hunnivant or helps him with his dilemma influences subsequent results, including the difficulty level of the game's final stage. Other side quests introduce Lloyd's parents (who themselves had traveled through the fog and met each other Cypress Laboratory) and involve Lloyd's attempt to save a neighbor family who had contributed to his life.

As the game's final stage approaches, the player faces a fork in the road. What should Lloyd do about the female voice, "Sophia," that has plagued him since his journey began? If the player chooses to listen to Sophia, she will guide Lloyd to the person responsible for the experiments and become a valuable ally. Her guidance makes game play easier for the player, and Sophia also describes the city's past and explains prior experiments that were considered failures. Alternatively, the player can choose to ignore Sophia and try to find the answers to Lloyd's questions without her help, providing a more challenging game experience.

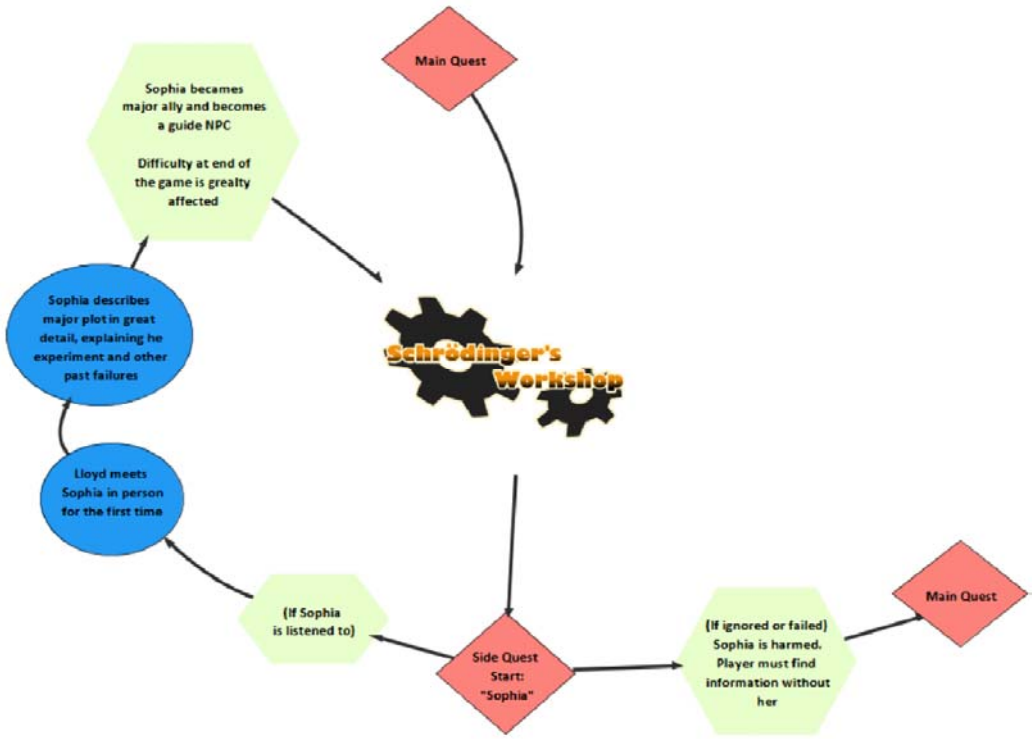

Figure A1. This is a student concept map that illustrates the aforementioned “Sofia” side quest.

Figure A2. This is the opening of the game. The camera will pan through Lloyd’s world as he carries out his normal day. The giant wall can be seen in the background of this image.

Figure A3. The “accident” occurs and technology comes to a halt. It is dark and smoke-filled, and the only light comes from the fires that spread throughout the city.

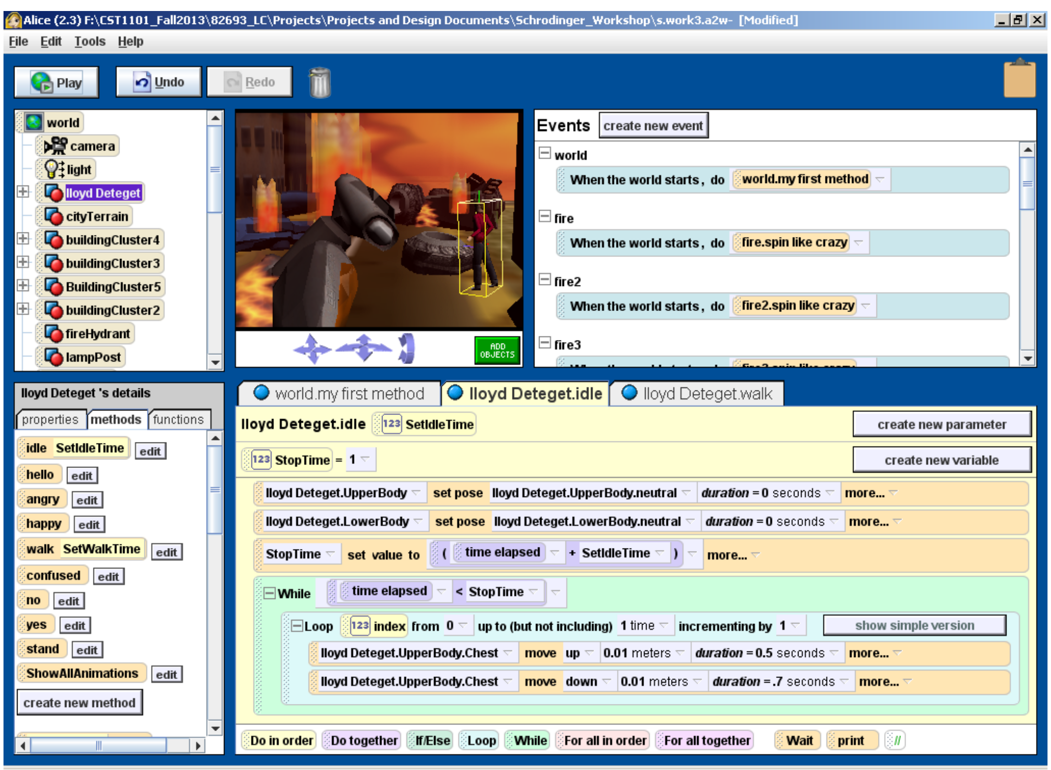

Figure A4. *Alice* screen shot showing the programming objects and methods used. Lloyd assesses damage up close. He is paralyzed and tries to reason with himself; this is Lloyd’s refusal of the call.

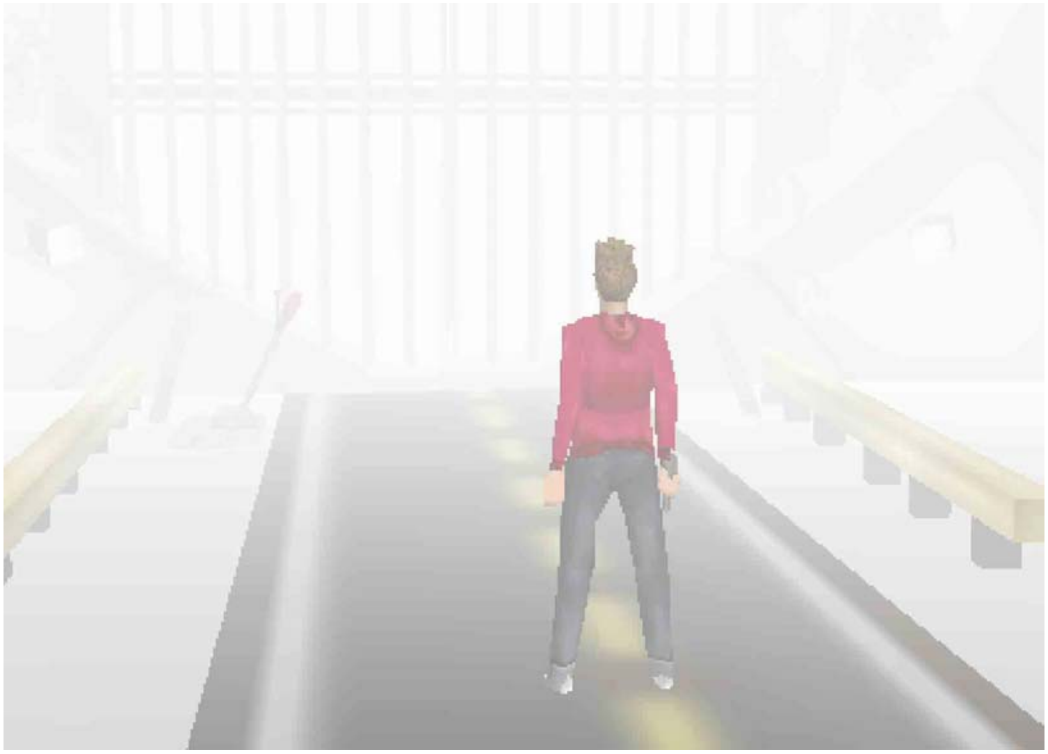

Figure A5. This image is the end of the prologue in the game. The student, Noel Melendez, used *Alice*'s fog function and varied subparts to create a convincing laboratory gate.