Composition and Computation: Integrated Learning via Video Games

Jim Brown, Becca Tarsa, Andrea Arpaci-Dusseau, Deidre Stuffer, University of Wisconsin-Madison
brownjr@wisc.edu, tarsa@wisc.edu, dusseau@cs.wisc.edu, stuffer@wisc.edu

Abstract: This Worked Example describes a three-course cluster offered during the 2012-13 academic year that uses video games to bridge the practices of written composition and computer programming. A group of 20 students enrolled in three courses—"Introduction to Computation," "Introduction to Composition," and a seminar course aimed at linking the concerns of the other two courses together—and spent the semester in an integrated learning environment that encouraged them to connect alphabetic writing with computer programming. By both analyzing and designing video games, these students were encouraged to see computational artifacts as expressive and rhetorical. This cluster will be offered again in the 2013-14 academic year. This Worked Example offers this course cluster as a way of using games to approach integrated learning projects, and it opens up a discussion about how to best use games in interdisciplinary teaching situations.

Introduction: Using Games to Study Composition and Computation

The idea for our example was to build a set of three integrated courses that give students the opportunity to work on writing skills alongside computer programming skills and to do so by having them create and critique video games (1). We carried out this experiment at the University of Wisconsin-Madison during the 2012-2013 academic year with an integrated cluster of three courses. The University offers a program for instructors who want to design such integrated learning communities, and the program is designed to help first-year students to transition into college life. Small groups of students (usually about 20) enroll in a three-course cluster that is organized around a common theme. One of the courses is a seminar that encourages students to see the interdisciplinary links between their work in the other two courses. In our cluster, students enrolled in "Introduction to Composition," "Introduction to Computation," and the seminar class (which was taught by both an instructor of record and a teaching assistant)

The idea of our cluster was to help students to see computer programming as a writing practice and traditional writing as a procedural (or even computational) enterprise. By exploring the links between both of these practices, we hoped to cultivate a sense of computational thinking that students could take beyond the classroom. Students were encouraged to understand traditional literacy practices alongside procedural literacy practices in a manner that emphasized the conceptual relationship between the two. By doing so, we hope to deepen students' awareness of computational and procedural forms as compositional and rhetorical practices.

Video games were one of the primary forms through which students explored these ideas and concepts. While students created various types of texts and computational artifacts, their primary projects in the seminar course involved the creation of video games that used what Ian Bogost (2010) calls procedural rhetoric, the use of computational processes in the interest of persuasion. Students created various games in "Introduction to Computation," wrote documentation for games and used other multimodal technologies to explain games in their "Introduction to Composition" class, and both critiqued and created video games in the seminar course. For instance, in the seminar they played games such as Molleindustria's *The McDonald's Game* and mtvU's *Darfur is Dying*, games that use computational procedures to make political arguments. Students analyzed how these games made arguments and then made their own games. Student games were inspired by Hal Ableson, Ken Ledeen, and Harry Lewis's *Blown to Bits* (2008), a book that they read in each of the three courses. By both writing and coding arguments inspired by *Blown to Bits*, students used all three courses to explore how games and other digital media open up different possibilities for communication and persuasion (2).

Inspirations and Motivations

This project takes much of its inspiration from recent explorations, in both composition and rhetoric and computer science, of the conceptual, creative and rhetorical similarities between programming and writing. Scholars such as Stuart Selber (2004), Andrea diSessa (2001), Bogost, Jeannette Wing (2006), and Cathy Davidson (2012) increasingly advocate movement towards an integrated pedagogy that diminishes the conceptual divide between writing a program and writing a paper.

Our central goal was to use analysis and creation of video games as an effective and engaging anchor for integrated learning, and we also had the additional goal, for this first iteration, of determining whether or not it was worth repeating – and, if so, what areas to focus the most energy on revising before next fall. We also hoped to get some insight into the best way to promote the cluster to students and administrators.

In addition to these underlying inspirations, all four instructors were inspired to participate in the project by research and pedagogical questions we've been mulling over individually in some form or another.

Jim Brown - Seminar Instructor: My research interests are in new media studies and software studies, and I am interested in giving students in humanities classes the opportunity to explore the rhetorical and literary possibilities of computation. I designed this cluster with the hopes that students could use my seminar to analyze and create computational artifacts that were not necessarily tools but were rather expressive artifacts. Video games were a particularly useful way to take up this work, and students in the course sought out ways to understand how their work in computer programming and composition classes could be combined in the interest of using computation to communicate and persuade.

Deidre Stuffer - Seminar TA: As a literature Ph.D. student I never imagined that I would be a technical presence in any classroom, but as the "Writing and Coding" TA I had to be the programming expert. In the end, the cluster was a good fit for my research interests. In my seminar papers I had been considering how technology mediates the writing process, reflected at the level of syntax and form. My experience with HTML and Java coding often prompted me to consider how digital objects challenge the metaphors through which we perceive the world. "Writing and Coding" allowed me to watch the students grapple with writing code and compositions as embodied acts. Furthermore, it broadened my horizons to the rhetorical possibilities of new media. I hope to teach the composition course in next year's version of this cluster, where perhaps I will catch glimpses of how contact with computational logic can alter student writing.

Andrea Arpaci-Dusseau - "Introduction to Computation" Instructor: As a faculty member in Computer Sciences, I am interested in teaching a wide variety of students about the importance of computation in everyone's lives. My goals in developing "Introduction to Computation" were to create a course that appeals to both CS majors and non-majors; as such, the course focuses on the big ideas of CS (e.g., creativity, abstraction, data, algorithms, and systems) instead of solely programming. Students use Scratch for programming, and thus learn about traditional programming concepts while creating motivating projects in a non-intimidating programming environment. For the programming projects, students experimented with randomized art, interactive stories, educational games, music composition, and basic data processing, in addition to a larger more open-ended project of their own choosing.

Becca Tarsa - "Introduction to Composition" Instructor: Since taking my first course in digital composition, my primary research interest has been adapting composition instruction - particularly freshman English - to best prepare students for the rapidly changing landscape of contemporary writing. The cluster's use of video games as a link between traditional composition practices and those of coding struck me as a great way to spread the work of integrating digital and traditional writing across multiple courses. I loved the idea that students could talk about games in the seminar, write about them with me, and build their own in their computer science course, and I'd still have time to prepare them for the traditional college essays they'll inevitably face before graduating.

Questions Asked by this Pedagogical Experiment

Our cluster of courses attempted to address a number of questions about writing and programming pedagogy and about how games can be used to bridge these two activities. We list those questions here in the interest of opening up a discussion about our worked example. This list is not exhaustive, and we aim to invite other GLS scholars to add more to this list.

- How can games be used as a bridge between computational and written literacies?
- How do we go about making useful connections between coded and written composition?
- How can we help students use written composition practices and assignments to deepen their procedural and computational literacy and vice versa?
- What kinds of assignments, cross-course collaborations, class formats, etc, best promote the goals of an integrated learning environment?
- And from a pragmatic standpoint: What should we be teaching these students, most of whom hope to

- enter computer science-related fields, to best prepare them for success in both their college and professional careers?

Scaffolding the Course Cluster

In the weeks leading up to this fall 2012 course cluster, the instructors discussed plans for each course both via email and in person. This gave us a sense of each other's individual plans for meeting the cluster's objectives, as well as general sense of the goals unique to each class. This helped us build a picture, as the semester progressed, of where the activities in our own classrooms overlapped with or built on those of the other two. One of the most tangible results of this front-end collaboration was the decision to use *Blown to Bits* as a common text across the three classes.

The text was already slated for use in the "Introduction to Computation" class, and Becca and Jim decided to integrate the text into their classes as well. This kind of collaboration is one important advantage of the course cluster format, which encourages collaborative course design. While each course incorporated other texts as well, *Blown to Bits* served as a touchstone across all three courses. The book fit well into all three courses since it explores the cultural implications of computation, and all four of us were excited about using it. We reasoned that having a shared text would help students to conceptually integrate their work across the three courses

Based on these collaborative preparations, Becca and Jim created syllabi aimed at speaking to the cluster's objectives while also meeting the goals of the individual course. (Not expecting to be part of the project until the last weeks of summer, Andrea had already finished her syllabus for "Introduction to Computation.") The Composition course used the 3-sequence model required by the University's first-year writing program by incorporating readings and assignments that connected the objectives of each sequence - reflection, research synthesis, and argument - to computational literacy as well as alphabetical literacy. Rather than writing traditional literacy narratives, for example, students were asked to reflect on the ways in which interaction with digital objects and environments has shaped their view of literacy. Becca designed a syllabus with the idea that a major game design project would serve as the central way for students to link their work in the other two courses.

Throughout the semester, students used the Scratch programming language to build a game that made a procedural argument about one of the chapters in *Blown to Bits*. Students worked in groups to build these games. They ran user tests, released multiple versions, and incorporated Picoboards (sensorboards designed for use with Scratch) to explore how games could be extended by way of physical computing. This semester-long group game design project served as the primary work of the course. Deidre prepared for her role by learning Scratch, the programming language students used for most of their coding assignments in both "Introduction to Computation" and the seminar. She designed a game of her own (*Bit Breaker*, a game that makes an argument about the persistence of digital information and issues of privacy) as an example. By coding, revising, and then presenting her game, Deidre showed students some of the possibilities of the Scratch platform.

Lessons Learned

We have completed one semester of this cluster, and it is again slated to be offered in 2013-2014. We are still awaiting course evaluations, and we hope to draw from these as we revise the courses for next year. While some of the instructors in this cluster will change, we do hope to carry over much of what we learned into the next version of the course.

This course cluster provided us with an opportunity to learn the advantages of integrated course design, and we learned that (with fairly minimal coordination) course clusters like this can be extremely effective at showing students that learning is not a siloed activity. Further, we learned that video games can be a particularly useful medium for exploring the links between different media literacies. While we faced the challenges that any educator using games challenges (for instance, students professing no interest in games or imagining that games are primarily for "fun" and are not "serious" academic work), we believe that the cluster's focus on both analyzing and producing games presented a convincing argument to students that games are a complex, rhetorically expressive medium.

In future iterations, we may use the seminar to have more theoretical conversations about the links between video games, composition, computation. While the seminar did provide some space for these discussions by examining Daniel Hillis' book *The Pattern on the Stone* (1999), the seminar was primarily focused on the video game design project. While we will not drop this focus on production (since it is so crucial for students to see the links between traditional writing and computer programming), the seminar might benefit from more balance between the theoreti-

cal and practical concerns of writing and coding. One way forward in this regard would be a focus on the burgeoning field of software studies, which uses the tools of humanistic analysis to examine software, video games, and digital fictions. One text that we will consider incorporating is the recently published *10 PRINT CHR$(205.5+RND(1)); : GOTO 10* (Montfort et al., 2012), a collaboratively authored book that examines a single line of code and "uses it as a lens through which to consider the phenomenon of creative computing and the way computer programs exist in culture" ("10 PRINT Website," n.d.). A text like this one would offer a specific example and would open up more questions about how computation and game design are threaded through culture and history.

Unexpected Challenges

As mentioned briefly above, we ran into the expected challenge of answering student resistance to the idea of video games as a meaningful and worthwhile learning focus. While many students (particularly those who were already avid gamers as well as aspiring CS majors) were an easy sell in this regard, those without preexisting interest in gaming or in computer science as a major were skeptical; this made it harder to engage them in the kind of integrated conceptual thinking about their work we wanted to foster.

It was also challenging to integrate assignments and course calendars across the three classes as fully and effectively as we'd initially imagined. While the time spent discussing and planning together before the semester began did facilitate some productive crossover between the courses, more in-depth coordination between course assignments proved more difficult than expected. For example, Becca had originally planned to have students produce documentation and other written support materials for the games they'd designed in the seminar class as their final project; however, the need to cover all the elements required by the first-year writing program made this logistically impossible to fit into the time remaining. An integrated project definitely seems feasible for future iterations of the course, but will require a bit more advance planning to make sure that it can be fit in around the program objectives, or designed to complement them as well as those of the cluster.

Assessing the Experiment and Soliciting Feedback

It can be tough to measure the success of long-term goals such as professional readiness and academic success without waiting a few years and tracking students down to follow up; obviously we won't be able to do that for some time, and realistically maybe not ever.

That said, there are still some useful measures of how we're doing. Most straightforwardly, if the cluster becomes an annual offering, as is our hope, enrollment statistics will also help us judge how things are going by showing us not only how many students are interested, but also who. Student evaluations are also a valuable tool here, especially with some effort on instructors' parts to encourage honest, reflective feedback. Further, the University program that sponsored this effort conducts their own assessment and shares it with instructors.

Just as important (if not more so) is how students perform within the courses - the quality of their work, their growth across the semester, the evidence in their discussions and projects of the kind of conceptual integration we're trying for. In the future, we hope to build some concrete ways of assessing this effort by way of qualitative and quantitative analysis.

Though there is always room for further improvement, there were definite signs of success in fostering integrated thinking about writing and coding, and about the links between computational and traditional literacy. In the composition class, well over half the students chose to pursue research topics related to concepts and topics they'd encountered in the other two courses. Many of these papers used video games as the focus through which to explore the concepts in question, suggesting that the course cluster's attempts to use games as a bridge between computing and writing were in fact successful. Students also frequently cited work from their other classes in the portfolios they compiled for their seminar class – further evidence that they'd begun to think about both written and coded work as forms of composition. On a simpler level, by the end of the semester students seemed very comfortable viewing all three classes as integrated learning experiences; references to readings or discussions from the other classes became increasingly common in all classes, and Jim even made a cameo appearance in several student projects for the other two courses.

The cluster was also a success in producing some excellent and interesting student work. Becca observed greater growth than is typical across student writing and overall performance throughout the semester, particularly in oral communication – students' end-of-semester presentations were exceptionally high quality, and delivered very comfortably and effectively. Student video games developed in both the "Introduction to Computation" course and the seminar were complex computational artifacts that used programming to make interesting political arguments about Internet privacy, data encryption, and copyright laws. This type of work likely stemmed more from

the increased sense of community created by having three classes together (two of them seminars) than the course cluster's objectives; however, it supports the idea of clustered courses as an effective and beneficial means through which to pursue those objectives.

While we are pleased with the first iteration of our Worked Example, we know that there are more avenues to explore. Through continued assessment and a dialogue with other scholars working at the intersection of game studies and pedagogy, we hope to continue to revise and improve this course cluster.

Endnotes

(1) The format for this Worked Example is loosely based on the Working Examples website (http://workingexamples.org/). We have used the questions on that site to guide our discussion, and we see this example as being in the "sprout" stage, given that we will be repeating the course cluster during the 2013-14 academic year.

References

10 PRINT CHR$(205.5+RND(1)); : GOTO 10. (n.d.).*10 PRINT CHR$(205.5+RND(1)); : GOTO 10*. Retrieved January 30, 2013, from http://10print.org

Abelson, H., Ledeen, K., & Lewis, H. (2008). *Blown to Bits: Your Life, Liberty, and Happiness After the Digital Explosion* (1st ed.). Addison-Wesley Professional.

Bogost, I. (2010). *Persuasive Games: The Expressive Power of Videogames*. The MIT Press.

Davidson, C. (2012, January 25). Why We Need a 4th R: Reading, wRiting, aRithmetic, algoRithms. *DMLCentral*. Retrieved July 17, 2012, from http://dmlcentral.net/blog/cathy-davidson/why-we-need-4th-r-reading-writing-arithmetic-algorithms

diSessa, A. (2001). *Changing Minds: Computers, Learning, and Literacy*. A Bradford Book.

Hillis, W. D. (1999). *The Pattern On The Stone: The Simple Ideas That Make Computers Work*. Basic Books.

Montfort, N., Baudoin, P., Bell, J., Bogost, I., Douglass, J., Marino, M. C., Mateas, M., et al. (2012). *10 PRINT CHR$(205.5+RND(1)); : GOTO 10*. The MIT Press.

Selber, Stuart (2004). *Multiliteracies for a Digital Age* (1st ed.). Southern Illinois University Press.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.