

Effectiveness of using a video game to teach a course in mechanical engineering

B.D. Coller

Department of Mechanical Engineering
Northern Illinois University
DeKalb, Illinois 60115
USA

M.J. Scott

Department of Mechanical & Industrial Engineering
University of Illinois at Chicago
Chicago, Illinois 60607
USA

Abstract

One of the core courses in the undergraduate mechanical engineering curriculum has been completely redesigned. In the new numerical methods course, all assignments and learning experiences are built around a video/computer game. Students are given the task of writing computer programs to race a simulated car around a track. In doing so, students learn and implement numerical methods content. The design of the course, around a video game, is rooted in commonly accepted theories of how people learn. The article describes a study to assess the effectiveness of the video game-based course. Results show that students taking the game-based course, on average, spend roughly twice as much time, outside of class, on their course work. In a concept mapping exercise, students taking the game-based course demonstrate deeper learning compared to their counterparts taking traditional lecture/textbook-based numerical methods courses.

Keywords: Interactive learning environments; simulation; applications in mechanical engineering; post-secondary education; programming.

1 Introduction

At Northern Illinois University, as at many places, undergraduate mechanical engineering students must take a course in numerical methods. The goal of the course is to teach students the fundamentals of how to get a computer to perform calculations that are too difficult or too cumbersome to perform and check by hand. Major themes include discretization, iteration, sources of numerical error, and practical management of that error. These are the fundamentals that undergird most of the modern engineering and scientific computational tools that have become indispensable in contemporary practice.

Despite its importance, it is a challenge to make the material engaging for students. Although top selling textbooks attempt to make connections to engineering, the homework problems are typically superficial, unconvincing, and uninspiring (Coller 2007). As we report in this article, students, on average, rate their (traditional) numerical methods course as one of the least important classes in the curriculum.

In 2005, we began teaching a new numerical methods course using a video game as its central theme. The game anchors almost all class instruction, learning exercises, assignments, and class projects. The goal is to leverage those aspects of video games that make them so engaging to adolescents and young adults.

Details of how we developed and implemented the course are outlined elsewhere (Coller, 2007). Here, we primarily focus on comparing learning outcomes achieved students taking the game-based course and those taking more traditional lecture/textbook-based courses. We also look at the amount of time that students spend on their coursework and the importance that they attach to the work. In a companion study, Coller and Shernoff (2009) present a preliminary study on student engagement as they work on coursework for their game-based classes. Before presenting the results, though, we provide a rationale for the unorthodox teaching strategy.

2 Why a video game?

One should think of video games as a distinct medium or mode of conveying information. What distinguishes the game from other common educational media such as books or videos is the degree of interactivity. Video games require their “players” to respond to events occurring in a simulated world. The players’ actions then affect the way that the simulation unfolds. At the heart of a video game is a computational model of the simulated world. The computational model, however, does much more than determine the “physics” of the video game world. A video game model also provides a series of challenges to test the player. The progression of challenges is often embedded in a storyline that has elements of humor, suspense, or drama. The computational model defines what it means to succeed or to fail. When video game designers put these elements together in compelling ways, they are able to effectively leverage the extra dimensions of engagement within the medium.

Over the past few years, scholars have written a new wave of books and reports urging educators to look to video games for ideas on how to engage students in deep, meaningful learning experiences (e.g. Gee, 2003; Shaffer, 2006; Aldrich, 2005, Federation of American Scientists, 2006). This was not a completely new idea, however. Academics have studied the engaging power of video games from an education perspective since the 1980s, when the United States was in the midst of a “Pac-Man Fever” epidemic (e.g. Bowman, 1982; Malone, 1981). However, as video games have evolved dramatically over the past few decades, the thinking about how they can impact education has become more sophisticated.

The goal of video game designers has always been to engage players. In the 1980s, there was also an incentive to keep the games short in duration so that adolescents with pockets full of coins would keep feeding the machines. Now, video games are primarily sold as software for consumers to play on their home equipment. Video games today are much longer in duration. Since players can even freeze and save a game in progress to be resumed at a later time, the games can last tens or hundreds of hours. The games are no longer simple tests of eye-hand coordination and reflexes. The most successful games often guide their players through rich and complex problem solving processes every bit as complicated, if not more challenging, than the types of homework problems students encounter in their undergraduate engineering courses. Advances in graphics, networking, and computational technologies have provided unprecedented possibilities for immersion into simulated worlds using common off-the-shelf equipment.

While video games are becoming longer, harder, and requiring more strategic and cerebral investment than ever, they are also becoming more popular than ever. The Kaiser Family Foundation recently reported (Roberts, Foehr, & Rideout, 2005) that 83% of children between 8 and 18 years old had at least one video game console in their home; 31% have three or more. Children of all

racers, genders, and economic status within this age group spend considerable time playing games, 68 minutes per day on average. In September 2007, Microsoft's video game, *Halo 3*, earned \$170M during its first 24 hours on the market. For comparison, the movie box office record is held by *Spider Man 3* which took in \$151M in its 3-day opening weekend earlier that year.

2.1 Learning Principles Embedded in Games

So why is Johnny able to learn and master the intricate cause and effect relationships in the video game *Roller Coaster Tycoon*, but seemingly unable to grapple with the second law of thermodynamics? Spending hours playing such games suggests a number of answers (Gee 2003; Kelly, 2005):

1. When players begin a video game, they plunge into it. They have no need for a manual. The goals of the game are clear. "Players know why they are learning something, and there are plenty of opportunities to apply what they learn." "There is little ambiguity about why knowledge is powerful since the power can be put to use immediately." (Kelly, 2005, p.3) Feedback as to whether they are achieving the goals is immediate, abundant, and unambiguous. Players are able to achieve initial success fairly quickly, but challenges intensify progressively to keep players at the edge of their abilities. Therefore, time on task is neither mundanely repetitious nor overwhelmingly difficult.
2. To help players progress, the most successful video games establish environments that encourage active and critical learning and have incorporated, whether intentionally or by accident, the need for superior learning strategies into the play. Players/learners can take risks in a space where real-world consequences are lowered. Necessary knowledge and skills are *discovered* from the bottom up through direct experiences, in a cyclic process of probing, reflecting, hypothesizing, and testing. Information becomes available to players/learners at just the time they will be able to make sense of it and to use it.
3. Finally, video games offer their players escapism or fantasy, stirring the imagination with a sense of unlimited possibilities. Players become gnomes, F-16 pilots, world rulers, (or engineers?). They experience a virtual world from a compelling perspective that is integrally linked to successful completion of the game.

The learning principles embedded in good video games are not unique to video games. They have a direct correspondence to constructivist theories of learning, active learning, and metacognition (Bransford, Brown, & Cocking, 2000). The explanation for why Johnny is a video game wizard but can't pass his engineering exams may lie in the fact that these same learning strategies are often absent from the classroom.

2.2 Video Games for Training and Learning

Spurred by a National Academy of Sciences report (Zyda & Sheehan, 1997), the U.S. Department of Defense has teamed with the video game industry to create some of the most compelling instructional video games. The game *Full Spectrum Warrior*, for example, is being used to teach (real) soldiers to be flexible and adaptable to a broad range of operational/combat scenarios. At the same time, a limited version has been released to the public and has become a commercial success.

Video games designed primarily for learning and/or training have been classified as "Serious Games." (See www.seriousgames.org). As a video game, *Full Spectrum Warrior* succeeds for the reasons enumerated in Section 2.1. As an instructional video game, Gee (2005) argues that it succeeds for the same reasons, but he emphasizes a feature which he calls "authentic professionalism."

As the game teaches the player to be a professional soldier, it demands that the player think, value, and act like a soldier to succeed.

In a recent book, Shaffer (2006) frames this feature within the language of epistemology. Video games have the potential of placing students in simulated environments where they face authentic, open-ended challenges similar in nature to those faced by real-world professionals. Because it is a simulated environment, aspects can be tailored to ease students into the roles of professionals. The consequences of failure are dramatically lowered. Students have the freedom to experiment with multiple approaches and learn from their relative success. Of course, the meaning of success is specified by the rules of the game. By defining the rules carefully, so that creative strategies are rewarded over formulaic ones, one may foster a system of meaning-making which requires students to think, value, and act like professionals. Learning is increasingly recognized to be a social and situated endeavor as learners participate within a “community of practice”: a group of people who share a concern, passion, or interest in a joint enterprise, and regularly interact in order to do it as well as possible. A profession is a quintessential community of practice (Lave & Wenger, 1991; Wenger, 1999).

In contrast, the epistemic frame of the traditional school setting, Shaffer argues, is one for which (almost) all questions have right or wrong answers, and the answers can be found in an all-knowing textbook or instructor. To exhibit knowledge is to correctly answer a battery of small, self-contained, narrowly focused examination questions. Generally, this is *not* what is valued in the real world beyond school.

Over the past decade, there has been a small but growing number of educators and researchers who have been experimenting with video game-based or game-enhanced instruction (e.g. Squire, 2004; Virvou, Katsionis, & Manos, 2005; Barab, Thomas, Dodge, Carteaux, & Tuzin, 2005; Shaffer, 2005; Jenkins, Klopfer, Squire, & Tan, 2003; Dev, Montgomery, Senger, Heinrichs, Srivastava, & Waldron, 2002; Blunt, 2006). However, research into their effectiveness is spread thinly over a wide range of subject areas, age groups, and educational settings. In a recent meta-analysis of computer gaming and interactive simulations for learning, Vogel et al. (J. J. Vogel, D. S. Vogel, Canon-Bowers, Bowers, Muse, & Wright 2006) considered 248 studies for inclusion in their work. However, only 32 of the studies met their requirements of being sufficiently rigorous. Similarly, Mayo (2007, p. 34) writes

There are perhaps only a handful of solid studies that rigorously measure the learning outcomes of immersive games compared to other teaching methods. Of them, few tackle science and engineering as subject matter.

In two recent reviews, Mayo (2007, 2009) is only able to report on two groups of researchers creating games in physical science and engineering higher education. One group from North Dakota State University has created games for geology and cellular biology (e.g. McClean, Saini-Eidukat, Schwert, Slator, & White, 2001). The other group is ours. In the current paper, we present, for the first time, detailed learning outcomes for students taking our game-based numerical methods course.

3 A Game-Based Numerical Methods Course

In the Spring of 2005, we began teaching our undergraduate numerical methods class with a video game called *NIU-Torcs*. In Figure 1 we show screen shots of *NIU-Torcs*. It has much in common with the *Need for Speed* series and *Gran Turismo 4*, the second best selling video game of 2005. We built *NIU-Torcs* on top of an existing open-source video game called *Torcs* (www.torcs.org) available under the GNU Public License. *NIU-Torcs* borrows most of the graphics engine of *Torcs*. Among other enhancements, we have given the game a higher fidelity simulation of the car’s physics,



Figure 1: Screen shots from *NIU-Torcs*.

including the engine, transmission, differential, suspension, tire mechanics, and more. In creating *NIU-Torcs*, we sought to straddle the boundary between rigorous engineering simulation and an accessible video game that could guide students through engaging and authentic engineering problems.

At the beginning of the course, each student receives his or her own car which sits motionless on a track. Students do not have steering wheels, gearshifts, accelerator, or brake pedals to get the car to move. Instead, each student must write a C++ program that gives the car its driving commands: how much to step on the gas pedal; how much to step on the brake pedal; which gear the transmission should be in; and how much the steering wheel should be turned to the left or the right. The driving program queries from the simulation important information such as the car's distance from the center line of the track; the heading angle of the car relative to the local heading angle of the track; wheel rotation rates; and copious information about the track itself which students may use in computing their driving strategies. Students compile their driving programs into a library which is then linked to the main *NIU-Torcs* code. Then, students are able to see the fruit of their effort. The car simulation runs in real time, displaying the behavior of the car in full 3D graphics.

Getting the car to simply move and navigate its way around the track is a fairly simple task. We have invited high school students onto campus to play the game; they are able to do it, usually within an hour or two. Making the car move fast and nimbly without skidding off the road, however, is a challenge that takes nearly fifteen weeks to fully realize. Students calculate the optimal instants to shift gears, the fastest speeds at which the car can navigate each turn, the best moment to begin braking before entering turns, and many more aspects of driving the car at the edge of its capabilities. Students seek sources outside the video game to learn numerical root finding, solving systems of linear algebraic equations, differentiation, integration of functions and ordinary differential equations, curve fitting, and simple optimization. For a detailed description of the tasks, and the numerical techniques used to solve the tasks, the interested reader is encouraged to see Collier (2007). The semester climaxes with an open-ended project in which students form teams and participate in a friendly competition. The final project rewards technical acumen as well as creativity. A six-minute video highlighting the students' work is posted at the web site: www.ceet.niu.edu/faculty/collier/video.htm.

3.1 A Comparison of Learning Activities

To compare the type of learning activity that occurs in the game-based numerical methods course to those that take place in a traditional course (or at least a typical undergraduate textbook), we present the example of root finding. It is one of the most fundamental topics in any undergraduate numerical methods course.

3.1.1 Root finding, a textbook problem. Below, we have reproduced a typical homework homework problem from Rao (2002), one of most widely used textbooks in engineering numerical methods courses.

(Rao, 2002, Problem 2.6) The normal stress induced at the inner fiber of a torsional helical spring is given by

$$\sigma_i = \left\{ \frac{4C^2 - C - 1}{4C(C - 1)} \right\} \frac{Mc}{I},$$

where $I = \pi d^4/64$, $c = d/2$, $C = D/d$, M is the bending moment, D is the mean coil diameter and d is the wire diameter. Find the value of C that corresponds to a stress of $\sigma_i = 55 \times 10^3$ psi when $M = 5$ lb-in and $D = 0.1$ in.

When one examines publishers' web sites (e.g. Pearson Higher Education, 2009; McGraw-Hill, 2009) as well as the prefaces of the books themselves (Rao, 2002; Chapra & Canale, 2006), one finds that problems such as the one listed above are a primary selling point for numerical methods texts targeted at engineering courses. Such problems have clear engineering contexts, and many make connections to previous engineering coursework.

But is the normal stressed induced at the inner fiber of a torsional helical spring something that naturally inspires the imagination of 20-year-olds, even the engineers-to-be? Will finding the "correct" answer to the question tell them anything that that they naturally want to know? If the answer is no (as we suspect), then what is the purpose of the engineering context? Is it really any better than a generic math problem with no connection to engineering?

With luck, students working on the problem above will learn a numerical root finding technique. But, which technique? And what will they learn about it? Chapter 2 of Rao (2002) presents eight root finding techniques that can be used in a variety of circumstances. It turns out that any of the eight can be used to solve the torsional spring problem. The problem does not stipulate which technique to use. There is no value to choosing a technique that converges quickly, compared to one with slow convergence. The problem only needs to be solved once so there is little benefit to choosing a technique whose iterative process starts easily. In fact, there is no need to use any of the numerical methods covered by the textbook. Students may use a plotting package to solve it graphically or they may perform a manual search by punching numbers into a pocket calculator. They may find a canned routine that generates the root(s) without requiring any thought at all.

Coller (2007) describes these types of problems as "artificial engineering problems" Effectively, they are generic math problems disguised in an engineering context. These problems, and others which have no connection to engineering/science whatsoever, make up the bulk of problems in two of the best selling books geared toward engineering students.

3.1.2 The root of motivation within the video game. We designed the game-based numerical methods course in a way that mirrors the design of good video game. In the first week, the goal is to get students hooked into the course project. At this moment, there is no direct connection to the core numerical methods content. We simply get students involved by challenging them to devise simple algorithms for steering the car toward the center of the road as it drives around a serpentine track. The task is not trivial. To figure it out, students must think deeply about how they keep their own car (or bicycle or tricycle) on a desired trajectory. Then they must encode the scheme into a short computer program. It almost never works correctly on the first try. But students are able to slow down the simulation and carefully compare what their algorithm is doing against what they think it should be doing at each instant of time. In relatively short time, students are able (sometimes with some prompting) to get their cars to drive a complete lap around the practice track, albeit slowly.

For most students, this is their first experience writing a computer program that directly determines how a machine acts and behaves. Previously, they had spent a semester learning elementary programming by writing programs that sort generic lists of names, and performs simple calculations on collections of numbers. By their nature, mechanical engineers like to build machines and to make them work. Although the cars that students drive with their computer programs are virtual, the algorithms they develop are real. The process of deriving the driving strategies is authentic.

In the first assignment, the car remains in first gear and reaches a top speed of about 50 mph (80 Km/h). After completing the first task, students begin exploring other possibilities, and we give them time to do so. They begin writing computer code to change gears, speed up in the straights, slow down for the turns, et cetera. It is an opportunity for students to begin the cycle of “hypothesizing, probing, and reflecting” that is an important component of successful video games as well as successful learning (Gee, 2002). Left to their own devices, though, the novice students create a tangled mess of computer code that does a mediocre job at driving on the practice track and is completely unable to adapt to different tracks, different pavement conditions, and different cars. Through direct experience, students recognize the need for a systematic, efficient, and rigorous approach to create driving algorithms. The educational objective of these initial activities is to motivate students, to make them eager to invest the time and effort to learn the computational methods that will allow them to dramatically improve their performance in the game.

3.1.3 A root finding problem within the game. A few weeks into the semester, students encounter a particularly challenging “level” in the video game. Students’ cars are placed on a long straight section of track, 700 meters from the finish line. Starting from a complete stop, the students’ computerized drivers must bring the car up to speed quickly and cross the finish line within a specified amount of time in order to successfully complete the event.

Since there are no turns in this portion of track, the strategy seems simple: just go as fast as possible. Giving a full throttle command to the virtual gas pedal is easy enough. However, in order to cross the finish line within the allotted time, students must program their drivers to shift gears from first gear through fourth gear at (almost) exactly the right moments.

Over a duration of two lecture periods, the students and instructor work together to formulate a strategy for calculating the optimal shift points. To summarize, students would drive their cars on an oval track with long straight sections to collect acceleration data. Figure 2a shows the track, while Figure 2b depicts acceleration versus speed data when the small sports car is in full throttle in each of the four gears. Upon examination of the plots, the optimal gear shifting strategy becomes evident: at any given speed, the driver should place the transmission in the gear which produces the largest possible acceleration. Thus the driver should shift from first gear to second gear at the speed for which the first and second gear acceleration curves intersect. Similarly, the optimal shift points from second to third gear and from third gear to fourth gear occur where the corresponding acceleration curves intersect. *Determining these intersection points is a root finding problem.*

3.1.4 A different kind of root finding problem. The root finding problem embedded within *NIU-Torcs* is different from the textbook problem in several respects. First, it arises naturally and authentically through an engineering problem that students have been working on since the beginning of the semester. And if the class/game was effective at motivating students that first week, then it is a problem that students *want* to solve. Second, the root finding problem is not a simple self-contained problem that fits conveniently within the confines of a single topic within the course. Like real-world engineering problems, achieving the goal requires students to conquer several technical hurdles and then piece together the facets into a properly functioning whole. In particular, notice that the root finding formulation requires students to obtain acceleration data versus speed for each gear. However, the software interface does not allow students to query the

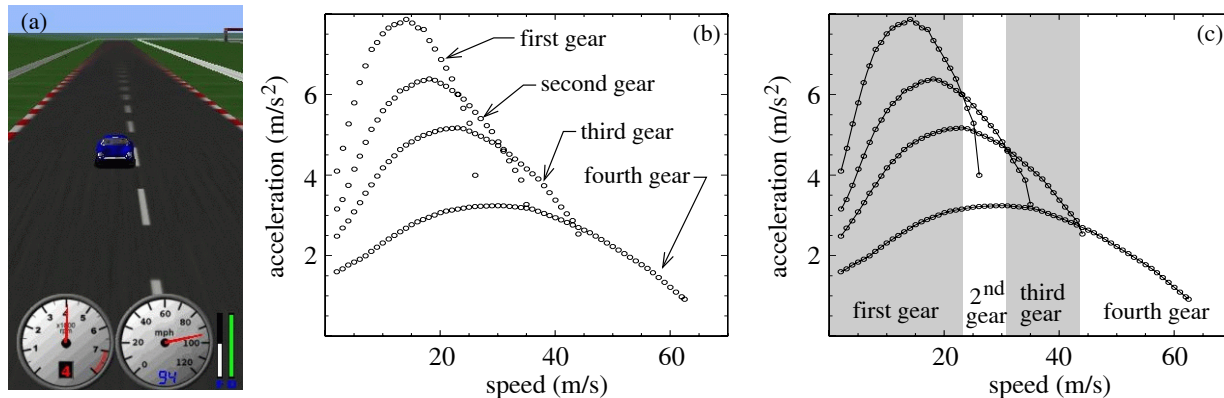


Figure 2: To determine optimal shift points, students must calculate speeds at which acceleration curves intersect. It is an authentic root finding problem.

speed or acceleration of the car directly from the simulation. They do have access to rotation rates for each of the four wheels (which rotate at different rates from each other depending on the braking/acceleration/cornering state of the car.) Students make some engineering judgments and then program a speedometer for their car. Next students must learn how to approximate derivatives of discretely sampled data (another numerical methods topic) in order to estimate acceleration. Furthermore, all root finding routines require continuous representations of functions. Therefore students needed to curve fit the discrete acceleration data (another numerical methods topic).

When it was finally time to perform the root finding, choice of numerical techniques was critically important. Any technique that relied on taking derivatives was doomed to fail: differentiation of discrete data is inherently noisy. Furthermore, students needed a technique that had robust convergence properties. Their shift point calculation methods were supposed to work with any car and any transmission, information students did not know a priori.

In summary, students had to make value judgments that arise naturally out of the problem. This is what happens in engineering practice. Students learned to think, act, and value as engineers do. They took on identities of engineers rather than mere engineering students. As such, the game is used to create a strikingly different type of learning environment compared to that of the textbook.

4 Effect of the Game-Based Course: Time on Task

Quoting from Bransford et al. (2000, pp. 57, 58), “In all domains of learning, the development of expertise occurs only with major investments of time, and the amount of time it takes to learn material is roughly proportion to the amount of material being learned.”

Given that young adults and teenagers, on average, spend considerable time playing video games, we were curious to see whether students taking the game-based numerical methods class spend more time on their coursework than students in traditional engineering courses. In the spring of 2005, we periodically surveyed students taking all core undergraduate mechanical engineering courses at Northern Illinois University (NIU). Among other questions, we asked students how much time per week they were spending, outside of formal lecture and lab, on their coursework.

We calculated averages for each class being offered, and then averaged over all courses being offered. Results are summarized in Figure 3. Each bar in the figure represents the average number of out-of-class hours per week reported by students in a class. The times are normalized by the average

over all courses. The figure clearly shows that students in the game-based numerical methods course spend more time on their coursework than all other courses in the curriculum. The value of 2.07 indicates that students in the game-based course spend roughly twice the average amount of time.

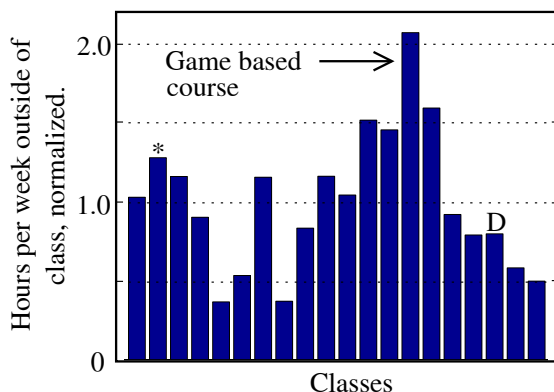


Figure 3: Average number of hours students spend outside of class each week on work for each core mechanical engineering class in the undergraduate curriculum. Hours are normalized by the average over all courses. The instructor who taught the game-based numerical methods course also taught the course indicated by the asterisk (*). The symbol ‘D’ denotes the senior capstone design course.

T-tests indicate that the differences between the game-based course and all other courses are statistically significant with $p < 0.001$. The game-based numerical methods course captured significantly more out-of-class hours than the experimental methods course, all technical electives, and the senior capstone design course (labeled with the letter ‘D’ in Figure 3).

There may be many factors that influence the amount of time students devote to a particular course, and the time surveys did not attempt to discern any cause and affect relationships. Yet, the results are consistent with those of Coller & Shernoff (2009) who present evidence from surveys conducted in 2007 that students working on game-based homework exhibit more “intellectual intensity,” more “intrinsic motivation,” and more “engagement,” compared to other mechanical engineering homework. Bransford et al. (2000, p. 77) state, “Students are motivated to spend the time needed to learn complex subjects and to solve problems that they find interesting.”

Despite the above-average work load, more than 90% of the eligible students taking the game-based numerical methods class later *chose* to sign up for the same type of work in an elective course in the same sequence within the curriculum.

5 Effect of the Game-Based Course: Learning Measures

Students in the game-based course invest more time in their course work. Does this translate to improved learning?

It is difficult to compare the degree of learning that takes place in the game-based course with that of a traditional numerical methods course. Different instructors emphasize different topics and assign work that provides different types of experiences. Students in the game-based course never even take (or study for) a midterm or final exam. Devising a test that does not favor a certain type of coverage is a considerable challenge.

5.1 Concept Map

Instead of having students answer questions on topics of our choosing, we had students tell us what they know about numerical methods by generating concept maps on the subject.

A concept map is a graphical construction consisting of nodes and lines. The nodes are terms corresponding to important concepts in the domain. The lines, sometimes accompanied by descriptive phrases, indicate relationships between concepts. For reference, we reproduce a portion of a student’s concept map in Figure 4.

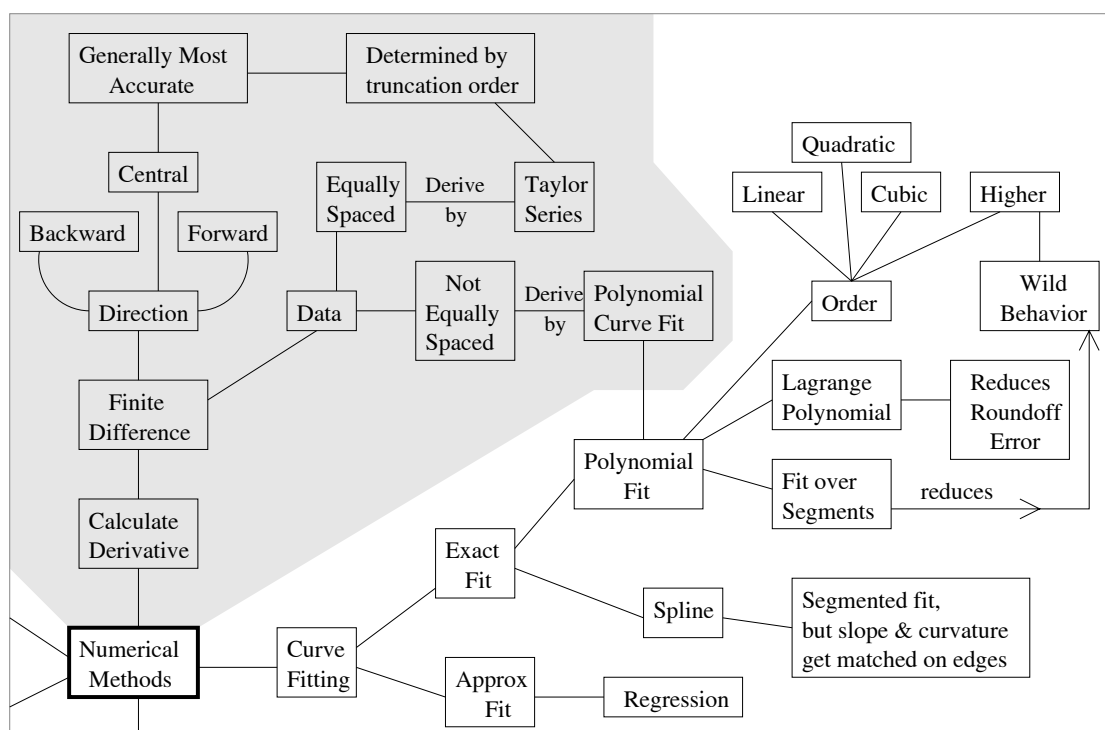


Figure 4: A portion of a concept map produced by a student for the subject “Numerical Methods.”

Before participating in the exercise, very few students had any experience creating concept maps. Therefore, we provided a training session for all participating students in which we presented a detailed example of a concept map on a neutral topic. Also, students received instructions, both orally and in writing, describing a series of steps for constructing the maps. Students were allowed to refer to their instructions and the example concept map as they were creating their own concept maps. A copy of the instructions is provided in the expanded on-line version of this article which may be found at www.ceet.niu.edu/faculty/coller.

In the first step of the instructions, we asked students to take a separate sheet of paper and quickly write down a list of words representing “sub-topics, concepts, actions, facts, and observations related to the main topic,” numerical methods. Students were told to keep writing until they slow down significantly.

In step two, we asked students to look for relationships between items on their list; they were told to pick out the main topics and themes, and to begin organizing the items hierarchically.

In the next step, we asked students to begin constructing the concept map. The instructions told students to write the words “Numerical Methods” in the center of a large blank sheet of paper and to draw a box around it. Using the example concept map as a guide, we explicitly told students to

write down the major topics onto their maps and to draw connections to the “Numerical Methods” block. Then we told them to insert sub-topics and sub-sub-topics into the map, making appropriate connections to the parent topics. The result is a dendritic representation of their understanding of “Numerical Methods.”

In the instructions, students were asked to grow and refine their trees. They were told to include anything that they directly or indirectly learned in the course, including the numerical techniques themselves, applications, and the ways in which they implemented the techniques. Students were told (verbally and in writing) that the map will be used to assess their level of understanding of the subject: “long branches indicate a deep understanding of the subject;” “many branches and sub-branches indicate a broad understanding of the subject;” and “connections between branches may indicate deep insight.” Finally, we told students that quality of the connections are of utmost importance. Connections must be meaningful to be awarded credit.

5.2 Underpinnings of the Concept Map

The purpose of the concept map is to provide a glimpse into how knowledge of numerical methods is organized in students’ minds. Over the past several decades, cognitive scientists have been studying the differences between novices and experts in fields as diverse as chess (de Groot, 1966), electronics (Egan & Schwartz, 1979), computer programming (Erlich & Soloway, 1984; Jeffries, Turner, Polson, & Atwood, 1981), architecture (Akin, 1980), history (Wineburg, 1991), and physics (Simon & Simon, 1978; Larkin 1979; Chi, Glaser, & Rees, 1982). In most cases, evidence suggests that expertise is *not* determined by one’s generic memory skills or mental processing speed. Instead, expertise is evident in the way that knowledge is organized in the mind. Quoting from Bransford et al. (2000, p. 38), in their review of the research:

Physics experts appear to evoke sets of related equations, with recall of one equation activating related equations that are retrieved rapidly. Novices, in contrast, retrieve equations more equally spaced in time, suggesting a sequential search in memory. Experts appear to possess an efficient organization of knowledge with meaningful relations among related elements clustered into related units that are governed by underlying concepts and principles... Within this picture of expertise, “knowing more” means **(1) having more conceptual chunks in memory, (2) more relations or features defining each chunk, (3) more interrelations among the chunks, and (4) efficient methods for retrieving related chunks...**

In the quotation above, we added the bold-face type and enumeration of points we wish to highlight. For the remainder of this article, these points shall serve as our working definition of “Knowing More.” The conceptual chunks in memory, the relations and features defining each chunk, and the interrelations among chunks are precisely what students are asked to express in their concept maps. What we get are those chunks that students are able to retrieve efficiently in the 20 to 60 minutes students take to complete the exercise.

Based upon Ausubel’s theory (Ausubel, 1968), Novak and co-workers (e.g. Novak & Gowin, 1984) coined the term *concept map*. It was originally thought of as learning tool, but later viewed as an assessment instrument (Novak & Ridley, 1988). As such, concept maps exhibit construct validity: there is a “direct correspondence between the elicited performance and the organization of cognitive structure as predicted by theories of learning.” (Novak & Ridley, 1988, p. 15).

There have been several studies that have demonstrated the ability of concept maps to differentiate between groups of varying expertise in the content domain assessed. (Acton, Johnson, & Goldsmith, 1994; Lay-Dopyera & Beyerbach, 1983; Markham, Mintzes, & Jones, 1994; Mintzes,

Wandersee, & Novak, 1997; Wilson, 1994) Acton et al. (1994) have shown that, although there is considerable variation in concept maps of physics experts, the experts’ maps were more similar among them selves than a group of students’ maps. Other studies which demonstrate validity of concept map assessment include McLure et al. (McLure, Sonak, & Suen, 1999) and Wallace & Mintzes (1990).

There is a broad spectrum of assessment activities that fall under the umbrella of concept maps. Generally speaking, the less directed activities in which students are responsible for providing the map structure, the linking lines, and propositions are thought to produce the best measures of students’ knowledge structures. (Ruiz-Primo & Shavelson, 1996; Ruiz-Primo, Schultz, & Shavelson, 2001; Yin, Vanides, Ruiz-Primo, Ayala, & Shavelson, 2005)

Ruiz-Primo et al. (1996) point out that having students generate the concepts that go into the map can be problematic. Students in that study generated irrelevant concepts and then formed accurate but irrelevant relationships. As a results scores on the maps were artificially high.

In our study, we find it difficult to supply concepts to the game-based and traditional numerical methods students completing the mapping exercise without biasing one group over another. Section 5.4 discusses a scoring scheme that attempts to filter out irrelevant relationships.

5.3 Administering the Concept Map Assessment

Over a period of two and a half years, we recruited a total of 86 undergraduate mechanical engineering students to construct concept maps for their numerical methods courses. At the time they participated in the exercise, a little less than half of the students were enrolled in the game-based numerical methods course; the remaining students were taking a traditional lecture/textbook- based numerical methods course.

At Northern Illinois University (NIU), students generally have a choice whether to take the game-based or a traditional numerical methods course. To account for any bias due to this self-selection, we also recruited students at the University of Illinois at Chicago (UIC), who do not have a choice and are required to take a traditional course. In Table 1, we summarize participation of students who generated concept maps for this study.

Course Type:	GAME	TRADITIONAL			
Institution:	NIU	NIU		UIC	
Instructor:	A	B	C	D	E
Number of Participants:	38	9	6	20	13
		15		33	
		48			
Average GPA:	2.76	3.02		3.09	

Table 1: Breakdown of participants constructing concept maps.

Students were recruited from seven different numerical methods classes. The two game-based numerical methods courses were both taught by instructor “A” (the lead author of this article); therefore, their data are lumped together in the first column.

Participants taking a “traditional” numerical methods course came from five different classes, taught by four different instructors labeled “B”, “C”, “D”, and “E,” at the two universities. The 20

students credited to instructor “D” came from two different semesters of the course. All instructors were tenured professors, each with at least eight years of teaching experience at the college level. All instructors had taught numerical methods multiple times. It is worth noting that instructor “E” has received an extraordinarily large number of teaching commendations, honors, and awards. None of the instructors teaching the “traditional” courses were directly involved in this research project.

We attempted to make the conditions under which the groups of students produced their concept maps as equivalent as possible. All students completed their concept maps during the final two weeks of scheduled lectures for the semester, before students typically begin studying for their final exams. All students completed their concept maps in a familiar academic classroom while being proctored by a familiar faculty member. All students were told ahead of time that they would be doing something related to their numerical methods course, but they were told that it was not necessary to prepare. Students were not allowed to refer to class notes or a textbook.

Despite our efforts and desires to make the groups of students as equivalent as feasibly possible, there were some differences. While all student participants explicitly volunteered, instructors A and D permitted the concept mapping exercise to be conducted during the regularly scheduled lecture. This probably explains why participation rates in these courses were higher. Also, it might explain why the average grade point average (GPA) for students taking the game-based course is lower than that of their counterparts taking “traditional” numerical methods courses. The sample for the game-based course spanned almost the entire population of students. There may have been a participation bias in the traditional courses in which students with higher GPA’s, perhaps more confident in their skills, were more apt to volunteer for the exercise.

It is worth noting also that there are differences between the “traditional” numerical methods courses at the two different schools studied. At UIC, it is a senior-level (fourth year) course, whereas students at NIU typically take the numerical methods course (game-based or traditional) in the third year. Based on catalog descriptions, the course at UIC appears to place more emphasis on applications than the traditional course at NIU. Also, we note that students at UIC, on average, enter the university with a higher composite ACT score, compared to those at NIU: 25.4 versus 22.6 respectively.

5.4 Quantifying Concept Map Features

We used two experts to score all 86 concept maps. Both scorers are tenured mechanical engineering professors who conduct research in computational science/engineering. Both scorers have taught undergraduate numerical methods courses. One of the two scorers was not directly related to the project and therefore had no stake in the outcome.

Before evaluating any student maps, the two scorers created their own concept maps for “Numerical Methods” at a level appropriate for the undergraduate course. We did *not* use the two maps as master maps for comparison to student maps (Ruiz-Primo & Shavelson, 1996). Instead, the expert maps were used to generate an initial list of concepts and map features that would be awarded credit.

The list served as a rubric with which to score students’ concept maps. But it was not a static list. As the scorers evaluated more concept maps – particularly maps from students taught by different instructors – they added more concepts and features to the list. All additions to the list/rubric were agreed upon by the two scorers.

The only items on the list were those directly related to the main topic, “Numerical Methods.” All items were concepts and topics covered in standard generic numerical methods texts (e.g. Rao, 2002; Chapra & Canale, 2006). However, not all topics in standard numerical methods texts

were eligible to be on the list. For example, the scorers intentionally left off any platform specific implementation topics. All references to software packages such as Matlab and Maple were omitted. All references to computer programming and specific data structures were omitted. All references to class projects and the contexts within homework assignments were omitted. The objective was to score only those aspects of numerical methods that are directly related to the subject, regardless of the mode of instruction. Through this mechanism scorers were able to filter out irrelevant information that comes as a byproduct of having students generate their own list of concepts (Ruiz-Primo et al, 1996).

5.4.1 Scoring the sophistication of the concept maps. The advantage of using a concept map assessment is that it “taps into the learner’s cognitive structure” (Novak & Gowin, 1984, p. 40) and reveals the sophistication (or lack of sophistication) of those structures. To score the structural properties of a concept map, Novak & Gowin (1984) propose a four-level system in which one counts the number of valid propositions (connections between concepts), the number of valid hierarchies, the number of valid cross links, and the number of valid examples. Then, one generates a composite score by calculating a weighted sum of the components. For example, hierarchical information may be assigned a weight of five times that of propositions, and cross links may be assigned a weight of ten times that of the propositions. Novak & Gowin (1984, p. 107) justify the weightings based on the fact that hierarchies “signal progressive differentiation and integrative reconciliation of concept meanings,” and “[cross links] signal possibly important integrative reconciliations and may therefore be better indicators of meaningful learning than are hierarchical levels.”

In evaluating the numerical methods concept maps, we adopted a similar multi-level scoring scheme. We identified four different measures of knowledge that can be quantified within the maps, measures that have elements in common with Novak & Gowin’s categories, but which are more aligned with the quote from Bransford et al (2000) in Section 5.2, defining expertise. Rather than generate a composite score with somewhat arbitrary weightings, we shall report the individual measures separately.

Measure #1: Number of major topics listed. Our first and crudest measure of student learning is obtained by counting the number of major topics presented in each student’s concept map. In the portion of the example map shown in Figure 4, the major topics are “Calculate Derivative” and “Curve Fitting.” Typically, the major topics reside in the first or coarsest level of the hierarchy.

Again, not all valid low-level connections in the hierarchy get counted in Measure #1. Only those on the scorers’ list that were deemed directly related to numerical methods were awarded credit. From the 86 concept maps, there were a total of 11 major topics on the list.

Measure #2: Number of numerical techniques per major topic. Our next measure quantifies the number of valid and relevant connections within the second level of the hierarchy. Here, the scorers count the number of specific numerical techniques associated with the major topics. The student who created the example map of Figure 4 received credit for three numerical techniques (“Polynomial Fit”, “Spline”, and “Regression”) associated with the major topic “Curve Fitting.” Additionally she or he received 0.1 points extra credit for listing “Lagrange Polynomial” a specific type of polynomial fit. Similarly, the student received a total of 1.5 points credit under “Calculate Derivative” for listing the three directions in which finite differences can proceed. The scorers’ list/rubric lists all possible point combinations encountered so that scores were applied uniformly.

After adding up the scores for all the specific numerical techniques, we normalized by dividing by the total number of major topics in Measure #1. Naturally, we expect students who list many major topics to also list many techniques. By normalizing, we get a sense of depth beyond Measure #1.

Measure #3: Number of defining features per major topic. The third measure looks

deeper into the hierarchy for evidence of students' knowledge. To describe what we mean by "defining features," it is perhaps best to return to the example in Figure 4. Within the "Calculate Derivative" branch of concept map, there are several *defining features* worthy of credit. For example, the student received one point for noting that central difference schemes are generally more accurate. The student also received credit for noting that equally spaced data and unequally spaced data must be handled differently. The student received an additional point for taking the data spacing distinction farther by noting how the formulae are derived in each of the cases (0.5 points for each).

There are similar defining features in the "Curve Fitting" branch of Figure 4. The student earned a point for categorizing numerical techniques into ones that fit the data exactly and others that fit approximately. Another feature of curve fitting is that if one attempts to exactly fit 17 data points with a 16th degree polynomial, one typically finds that the curve oscillates erratically between the data points. The student makes this observation near the upper right corner of the concept map. Furthermore, the student explains how to fix this problem. By breaking the data into segments and fitting it segment-wise with low order polynomials, one can eliminate the "wild" oscillations. This observation earned another half point (before normalization) within Measure #3. Additionally the student received credit for one *feature* by distinguishing the spline from other methods. The block "Reduces Roundoff Error" connected to "Lagrange Polynomial" was awarded credit for only half a *feature*. Full credit would have been awarded if the student had provided some description of why or how the method reduces roundoff error.

As with the previous measure, we normalize the total number of features by the number of major topics to arrive at Measure #3.

Measure #4: Number of connections between primary topics. This final measure is equivalent to Novak & Gowin's (1984) "cross links" or connections between branches. In the brief training session, students were shown examples of inter-branch connections in the sample concept map. In the instructions, students were explicitly asked to make meaningful connections between branches.

Referring back to Figure 4, we see that the dominant topology is that of branching. We see just a few major topics emanating from "Numerical Methods." Then the main trunks of the concept maps split into numerical techniques. The numerical techniques branch into details of the techniques. As one gets farther from the central topic, the branches keep splitting, providing finer detail. This is typical.

However, there are a few instances in which the branches connect to other branches. In the top left portion of the map, for example, the branch "Data" → "Equally Spaced" → "[derived by] Taylor Series" is connected to the branch "Central" → "Generally Most Accurate" → "Determined by truncation order" → "Taylor Series." This type of connection between branches did *not* garner any credit according to the rubric for Measure #4. This type of interconnection is not sufficiently deep. In fact, there are ways of re-organizing the concepts in the "Finite Difference" branch so that the same information is presented *without* a topological connection between branches or sub-branches. Similarly, the connection at "Wild Behavior" on the right side of the concept map in Figure 4 was *not* deemed sufficiently rich to warrant credit for Measure #4.

The inter-concept connection in the example map that did receive credit lies at the center of Figure 4. When applying finite differences to unequally spaced data, one uses formulae derived by fitting the data with a polynomial and then take the derivative of the polynomial. The student who created the concept map recognized this and made an explicit connection to the curve fitting branch of the concept map. Observe that the the entire "Calculate Derivative" branch of the concept map is shaded, while the "Curve Fitting" branch is not shaded. This shading scheme highlights the fact that the latter interconnection links primary branches of the map.

There were 9 different types of interconnections for which students received Measure #4 credit.

All of them are connections between primary branches. One connection that the student who created the map in Figure 4 almost made, but missed, is a connection between polynomial curve fitting and a primary branch devoted to solution of systems of linear algebraic equations. When fitting a polynomial directly, one generates a system of linear algebraic equations. However, the procedure yields Vandermonde-type equations that are generally ill-conditioned and are susceptible to roundoff error. The student prescribes how to alleviate this roundoff error, but since she/he does not make explicit connection to linear algebra, no credit for Measure #4 is awarded.

In our calculation of Measure #4, we do not normalize the number of interconnections. In measures #2 and #3, we expect the number of numerical techniques and the number of defining features to scale roughly linearly with the number of main topics, thus justifying the normalization. The number of interconnections, however, does not scale in the same way.

5.4.2 Reliability of scoring. With the detailed rubric, carefully specifying how concept map features are evaluated, our scoring scheme has much in common with the “relational scoring” scheme which McLure et al. (1999) deemed most reliable and valid in their study of several different scoring techniques. Once the rubric is created, the scoring scheme places relatively little cognitive load on the evaluator.

In our case, the two scorers evaluated each of the concept maps independently and then compared notes for discrepancies. They had spreadsheets that listed all rubric items and automatically tallied scores for the four measures, so discrepancies were easy to pinpoint. On relatively rare cases discrepancies appeared, the two scorers reached consensus.

Following a comment from one of the reviewers of an early draft of this article, the two scorers reevaluated fifteen randomly selected concept maps. The reevaluations occurred more than a year after the original assessments. Yet, both evaluators reproduced the original the original scores, in all categories, over 98% of the time. In all cases, the errors were ones which an evaluator missed an item on the map. On the reevaluation, no item was missed by both evaluators. It is not uncommon for concept map studies to report such high inter-rater reliability (Ruiz-Primo & Shavelson, 1996).

5.5 Comparing Students in Game-Based and Traditional Courses

In Figure 5, we present results of the concept map scoring. For each measure, the average score of students taking the game-based course is displayed next to the average score of students taking traditional numerical methods courses. To the right side, we display the averages for the traditional classes taught by each of the four instructors. The four traditional classes are labeled “B,” “C,” “D,” and “E,” in correspondence to the instructor labels in Table 1.

For each of the four measures, we performed two independent sample t-test to assess the validity of the null hypothesis that the means of game-based and traditional students are the same. One test compares the average of the game-based students to the average of students in all the traditional courses. The other test compares game-based students to those in the highest scoring traditional class. The p -values of the statistical tests are reported in Figure 5.

5.5.1 Evidence of low-level knowledge. In Section 5.2, we established our working definition of “Knowing More,” based upon the recent book, *How People Learn* (Bransford et al., 2000), summarizing decades of research in cognitive science. Concept map measures #1 and #2 cover just the first dimension of that definition: “having more conceptual chunks in memory.” It is a relatively low level of learning; the major topics and their associated techniques are listed in the textbook’s table of contents.

In Figure 5, we see that all the scores in Measure #1 are very similar. With p -values of 0.298 and 0.849, we cannot claim any difference between the groups.

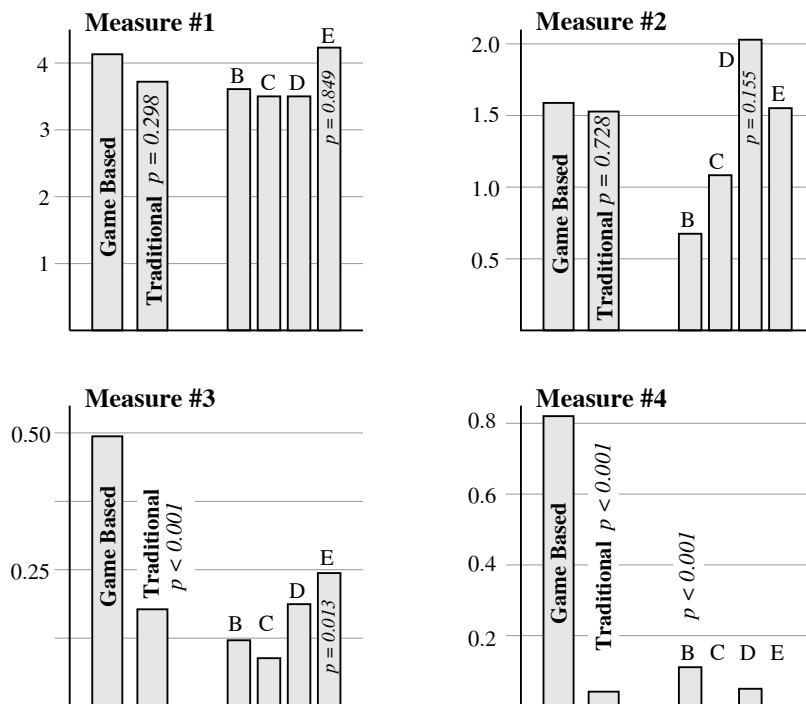


Figure 5: Averages for the four measures of student learning outlined in Section 5.4.

For Measure #2, we see more variability among the different instructors. However, the average of the game-based students and the traditional students on the whole are nearly identical. The average Measure #2 score for students under instructor “D” is considerably larger than that of the game-based students. However, with $p = 0.155$, we cannot say that the difference is statistically significant.

At the lowest levels of learning, we find no statistically significant differences between the game-based and traditional students.

5.5.2 Evidence of deeper learning. In asking students to complete concept maps, we invited them to tell us what *they* think is important within the topic “Numerical Methods.” Thus they had the opportunity, if they chose to take it (and they were encouraged to do so), to provide more details about the techniques they listed. In response, many students listed situations for which the methods might be or might not be appropriate. Several compared and contrasted computational efficiency of numerical routines. Some students discussed/compared sources of error. Students categorized iterative procedures, noted theoretical foundations, and more. These are precisely the high level features that Measure #3 captures.

Measure #3 is where we begin to see dramatic differences between student in the game-based course and those taking traditional numerical methods courses. The average score for the game-based students was more than two and a half times higher than their counterparts’ average score. The t-test yielded $p < 0.001$, indicating high statistical significance. While there was some noticeable variability among the traditional classes taught by the four instructors, the game-based average for Measure #3 is still twice as large as that for the best traditional course. With a p -value of 0.013, the difference observed between the game-based students and those in the highest scoring traditional course is significant.

In Measure #4, the game-based course average was 0.82. This means that, on average, students

in the course found 0.82 deep interconnections between concepts. While such interconnects were not abundant for the game-based students, we would categorize them as prevalent. More than half the students generated such interconnections. Of those, many students presented multiple interconnections.

In contrast, the interconnections were almost nonexistent among students taking traditional numerical methods courses. Only 2 of 48 students produced them. Thus the game-based students dramatically outperformed their counterparts in Measure #4.

Now, when we turn to the second and third components of our working definition of “Knowing More,” we find that the game based students have significantly “more relations or features defining each chunk,” and significantly “more interrelations among the chunks.” The two groups of students are quite different, as measured by our concept map assessment.

5.5.3 Other evidence of deep learning. While scoring the concept maps, we found some interesting connection structures which our measures were not able to quantify. Specifically, there was a group of students who overlaid a categorization scheme on top of the branching concept map. The general topology of this feature is depicted in Figure 6.

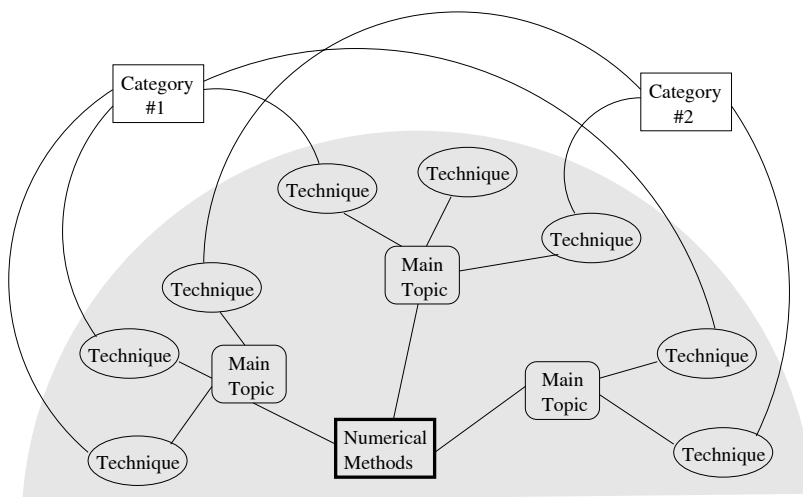


Figure 6: Schematic depiction of categorization in the numerical methods concept maps.

In the shaded part of the figure, we show a typical generic concept map structure. The feature details and interconnections captured by Measures #3 and #4 are omitted from the figure for clarity. Sitting on top of this structure is a classification into two categories. Almost all numerical techniques were then associated one category on the other. This way of forming the concept map is not particularly elegant. However, it conveys important information.

Two of the 86 students chose to categorize numerical according to those which require iteration and those which produce a precisely correct answer (up to numerical precision) in one step. Similarly, four students classified techniques into those which are prone to roundoff error and those which are prone to truncation error. All six of these students took the game-based numerical methods course.

Research of Chi et al. (1982), in the context of undergraduate physics education, points out that novice students, even though they were educated in a manner organized around physical principles, tend to classify canonical elementary physics problems according to their “dominant objects.” Novices, for example, lump together all inclined plane problems, regardless of the way in which one solves the problems. Experts, in contrast, categorize problems according to the physical principles (e.g. work-energy, conservation of momentum) best suited determine the solution. This distinction

is one of Chi et al.'s (1982) defining characteristics of expertise. The fact that all six students who provided this higher level categorization came from the game-based course provides further evidence that the students in the game-based course, on average, had deeper learning experiences.

6 Effect of the Game-Based Course: Perceived Importance

Finally, we surveyed groups of students within a few weeks before they graduated with their Bachelor of Science degree in Mechanical Engineering. We gave students a list of the 23 engineering and computer science courses required for their degree. They were asked to rank the courses on a scale from 1 (low) to 10 (high) indicating “how *important* [they] feel each course was in [their] education toward becoming a mechanical engineer.” Students were advised to evaluate the importance of the course material rather than the quality of instruction. To force students to discriminate in their ratings, they were not allowed to assign a single number to more than five courses.

We collected responses from 58 students, 22 of whom took the game-based numerical methods course. The remaining 36 took a traditional numerical methods course. Twenty of these took the traditional course before Spring 2005, when students did not have a choice (game-based vs. traditional) of which course to take.

6.1 Results

We normalized raw survey responses by individual to generate z-scores, so that each individual's distribution of responses has a mean of zero and unit standard deviation. Responses were therefore transformed to reflect the deviation from that individual's own mean on a standardized scale.

In reviewing the data, we found only two significant differences between the groups of students. Most significantly, students who took a traditional numerical numerical methods course, on average, rated its importance with a z-score of -0.58. Those who took the game-based numerical methods course rated it with a z-score of 0.76. The difference is significant with $p < 0.001$ on a standard two-tailed t-test.

Among students taking the traditional course, numerical methods was the lowest rated mechanical engineering course. Only a computer science course, an electrical engineering course, and an industrial engineering course rated lower. The addition of the video game (along with all the changes built around it) transformed the course from being one of the least important to one of the most important in the minds of students.

The other significant difference lies how students perceive the importance of their elementary computer programming course, a prerequisite to the numerical methods course. Students taking the game-based numerical methods course saw more importance in their initial programming course (z-score of 0.07 versus -0.65 for students taking the traditional numerical methods course).

7 Closing Remarks

When we began teaching the game-based numerical methods course, we sensed that there was something unusually good happening educationally. Students seemed more interested, more engaged, and more invested in learning the material. Objective evidence now supports our hunch. Students in the game-based course invest more time in their work. They view the course content as more valuable. Most importantly, evidence supports our claim that students taking the video game-based course learn the material at a deeper level than students taking typical traditional lecture and textbook-based courses. Our comparison group used as a control consists of students taking the course from four different instructors at two different but comparable institutions. We consider

them to be representative of the typical traditional numerical methods course. In our two deepest measures of learning, differences between game-based students and traditional students were considerably greater than differences between different instructors who taught the traditional numerical methods courses.

Our results are consistent with those in a companion paper (Coller & Shernoff, 2009) which reports preliminary results showing that students in the game based course are more engaged. Engagement in the study is measured by a technique called the Experience Sampling Method.

What is unclear is the degree to which the game itself is responsible for the deeper learning. To incorporate the video game, we had to completely re-develop the course. After all, to simply work the game into some homework assignments, yet still keep the class structure the same would be to fundamentally ignore what makes video games engaging (Gee, 2003). In essence, the course was *about* the video game and *about* writing driving algorithms that could squeeze the maximum possible performance out of a simulated car.

Using the terminology of Shaffer (2006), we changed the *epistemology* of the course. Traditional courses are often about doing homework and preparing for exams with the goal of getting a good grade. Success is usually measured by how many “correct answers” one is able to produce to a series of short, overly simplified, narrowly focused, disconnected problems. In contrast, the game-based course has an epistemology more aligned with that of a professional engineer. Students are given problems that are big, messy, interdependent, and above all, authentic. Because they are given genuine tasks with realistic constraints, and meaningful metrics for success, students think about and value the academic subject material the way a professional does. Technical details matter. The bean counting that goes into determining a grade becomes secondary.

Such learning environments, Shaffer argues (2006), are more motivating and more effective at developing deep, innovative thinkers. The video game itself might not be as important as the learning environment we built around it, though. It seems reasonable to assume that some other type of engaging, authentic (non video game) project could have served the same purpose. Video games are simply one convenient and effective place to attach an epistemic educational scaffolding.

Acknowledgment

The authors gratefully acknowledge the support from the National Science Foundation of the United States under grants 0354557 and 0633162. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect those of the National Science Foundation. The authors also acknowledge Ugo Buy and Francis Loth for their important roles in the project.

References

- Acton, W. H., Johnson, P. J., and Goldsmith, T. E. (1994). Structural knowledge assessment: Comparison of referent structures. *Journal of Educational Psychology*, 86:303 – 311.
- Akin, O. (1980). *Models of Architectural Knowledge*. Pion.
- Aldrich, C. (2005). *Learning by Doing*. Pfeiffer.
- Ausubel, D. P. (1968). *Educational Psychology: A Cognitive View*. Holt Rinehart and Winston.
- Barab, S., Thomas, M., Dodge, T., Carteaux, R., and Tuzin, H. (2005). Making learning fun: Quest Atlantis, a game without guns. *Educational Technology Research & Development*, 53:86 – 107.

- Blunt, R. (2006). *A Causal-Comparative Exploration of the Relationship Between Game-Based Learning and Academic Achievement: Teaching Management with Video Games*. PhD thesis, Walden University.
- Bowman, R. (1982). A “Pac-man” theory of motivation: tactical implications for classroom instruction. *Educational Technology*, 22(9):14 – 17.
- Bransford, J., Brown, A., and Cocking, R. (2000). *How People Learn*. National Academies Press.
- Chapra, S. and Canale, R. (2006). *Numerical Methods for Engineers*. McGraw-Hill, fifth edition.
- Chi, M., Glaser, R., and Rees, E. (1982). Expertise in problem solving. In Sternberg, R., editor, *Advances in the Psychology of Human Intelligence*, volume 1, pages 7 – 75. Lawrence Erlbaum Associates.
- Coller, B. (2007). Implementing a video game to teach principles of mechanical engineering. In *Proceedings of the 2007 ASEE Annual Conference and Exposition*.
- Coller, B. and Shernoff, D. (2009). Video game-based education in mechanical engineering: A look at student engagement. *International Journal of Engineering Education*, 25(2):308 – 317.
- de Groot, A. (1966). Perception and memory versus thought: Some old ideas and recent findings. In Kleinmuntz, B., editor, *Problem Solving*. Wiley.
- Dev, P., Montgomery, K., Senger, S., Heinrichs, W., Srivastava, S., and Waldron, K. (2002). Simulated medical learning environments on the Internet. *Journal of the American Medical Informatics Association*, 9:437 – 447.
- Egan, D. E. and Schwartz, B. (1979). Chunking in recall of symbolic drawings. *Memory & Cognition*, 7:149 – 158.
- Ehrlich, K. and Soloway, E. (1984). An empirical investigation of the tacit plan knowledge in programming. In Thomas, J. and Schneider, M., editors, *Human Factors in Computer Systems*, pages 113 – 134. Ablex.
- Federation of American Scientists (2006). Harnessing the power of video games for learning. Technical report, Federation of American Scientists.
- Gee, J. (2003). *What Video Games Have to Teach Us About Learning and Literacy*. Palgrave MacMillan.
- Gee, J. (2005). What would a state of the art instructional video game look like? *Innovate*, 1(6).
- Jeffries, R., Turner, A., Polson, P., and Atwood, M. (1981). Processes involved in designing software. In Anderson, J., editor, *Cognitive Skills and their Acquisition*. Lawrence Erlbaum Associates.
- Jenkins, H., Klopfer, E., Squire, K., and Tan, P. (2003). Entering the education arcade. *ACM Computers in Entertainment*, 1(1).
- Kelly, H. (2005). Pre-summit paper: Harnessing the power of games for learning. In *Summit on Educational Games*.
- Larkin, J. (1979). Information processing models in science instruction. In Lochhead, J. and Clement, J., editors, *Cognitive Process Instruction*, pages 109 – 118. Lawrence Erlbaum Associates.

- Lave, J. and Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press.
- Lay-Dopyera, M. and Beyerbach, B. (1983). Concept mapping for individual assessment. Technical report, School of Education, Syracuse University. ERIC Document Reproduction Service No. ED 229 399.
- Malone, T. (1981). Toward a theory of intrinsically motivating instruction. *Cognitive Science*, 4:333–369.
- Markham, K. M., Mintzes, J. J., and Jones, M. G. (1994). The concept map as a research and evaluation tool: Further evidence of validity. *Journal of Research in Science Teaching*, 31(1):91 – 101.
- Mayo, M. (2007). Games for science and engineering education. *Communications of the ACM*, 50(7):31 – 35.
- Mayo, M. J. (2009). Video games: A route to large-scale STEM education? *Science*, 323:79 – 81.
- McClean, P., Saini-Eidukat, B., Schwert, D., Slator, B., and White, A. (2001). Virtual worlds in large enrollment biology and geology classes significantly improve authentic learning. In Chambers, J., editor, *Selected Papers from the 12th International Conference on College Teaching and Learning (ICCTL-01)*, pages 111 – 118.
- McGraw-Hill (2009). Numerical Methods for Engineers by S.C. Chapra & R.P. Canale. Publisher web site: highered.mcgraw-hill.com/sites/007291873x/information_center_view0/. Retrieved April 2, 2009.
- McLure, J. R., Sonak, B., and Suen, H. K. (1999). Concept map assessment of classroom learning: Reliability, validity, and logistical practicality. *Journal of Research in Science Teaching*, 36(4):475 – 492.
- Mintzes, J. J., Wandersee, J. H., and Novak, J. D. (1997). *Teaching Science for Understanding*. Academic Press.
- Novak, J. D. and Gowin, D. B. (1984). *Learning How to Learn*. Cambridge University Press.
- Novak, J. D. and Ridley, D. R. (1988). Assessing student learning in light of how students learn. Technical report, American Association for Higher Education.
- Pearson Higher Education (2009). Applied Numerical Methods for Engineers and Scientists by S.S. Rao. Publisher web site: www.pearsonhighered.com/educator/academic/product/0,013089480X,00\%2Ben-USS_01DBC.html. Retrieved April 2, 2009.
- Roberts, D., Foehr, U., and Rideout, V. (2005). Generation m: Media in the lives of 8-18 year-olds. Technical report, Kaiser Family Foundation.
- Ruiz-Primo, M. A., Schultz, S. E., Li, M., and Shavelson, R. J. (2001). Comparison of the reliability and validity of scores from two concept-mapping techniques. *Journal of Research in Science Teaching*, 38(2):260 – 278.
- Ruiz-Primo, M. A., Schultz, S. E., and Shavelson, R. J. (1996). Concept-map based assessment in science: An exploratory study. In *Paper presented at the AERA Annual Meeting, Chicago, IL*.

- Ruiz-Primo, M. A. and Shavelson, R. J. (1996). Problems and issues in the use of concept maps in science assessment. *Journal of Research in Science Teaching*, 33(6):569 – 600.
- Shaffer, D. (2005). Epistemic games. *Innovate*, 1(6).
- Shaffer, D. (2006). *How Computer Games Help Children Learn*. Palgrave MacMillan.
- Simon, D. and Simon, H. (1978). Individual differences in solving physics problems. In Siegler, R., editor, *Children's Thinking*, pages 325 – 348. Lawrence Erlbaum Associates.
- Squire, K. (2004). *Replaying History*. PhD thesis, Indiana University.
- Virvou, M., Katsionis, G., and Manos, K. (2005). Combining software games with education: Evaluation of its educational effectiveness. *Educational Technology & Society*, 8(2):54 – 65.
- Vogel, J. J., Vogel, D. S., Canon-Bowers, J., Bowers, C. A., Muse, K., and Wright, M. (2006). Computer gaming and interactive simulations for learning: A meta-analysis. *Journal of Educational Computing Research*, 34(3):229 – 243.
- Wallace, J. D. and Mintzes, J. J. (1990). The concept map as a research tool: Exploring conceptual change in biology. *Journal of Research in Science Teaching*, 27(1):1033 – 1052.
- Wenger, E. (1999). *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press.
- Wilson, J. M. (1994). Network representations of knowledge about chemical equilibrium: Variations with achievement. *Journal of Research in Science Teaching*, 31(10):1133 – 1147.
- Wineburg, S. (1991). Historical problem solving: A study of the cognitive processes used in the evaluation of documentary and pictorial evidence. *Journal of Educational Psychology*, 83(1):73 – 87.
- Yin, Y., Vanides, J., Ruiz-Primo, M. A., Ayala, C. C., and Shavelson, R. J. (2005). Comparison of two concept mapping techniques: Implications for scoring, interpretation, and use. *Journal of Research in Science Teaching*, 42(2):166 – 184.
- Zyda, M. and Sheehan, J. (1997). *Modeling and Simulation: Linking Entertainment and Defense*. National Academies Press.

Appendix: Rubric for Scoring Concept Maps

For the first three measures, the list of “major topics,” “techniques,” and “defining features” reported by students is provided below. Except where noted, each item listed on the concept map was awarded one point in the corresponding measurement category.

Major Topics	Techniques	Defining Features
Number Representation	Binary Representation	Nature of roundoff.
	Binary Arithmetic	Speed of operations.
	Data Types	Manipulations to minimize roundoff.
Root Finding	Newton-Raphson	Convergence & Divergence.
	Bisection	Speed of convergence.
	Secant	Starting: bracketing vs. initial guess.
	Regula Falsi	No roots, multiple roots.
	Incremental Search	Explain how it works. (0.5)
Systems of Linear Algebraic Equations	Gauss Elimination (G.E.)	Iterative vs. direct approaches.
	Pivoting (assoc. w/ G.E.)	Pivoting for error control.
	Cramer’s rule	Pivoting to avoid divide by zero.
	LU Decomposition	Complexity analysis.
	Crout’s Method	Existence of solutions, determinants.
	Jacobi Iteration	Size & structure suggest technique.
	Gauss Seidel Iteration	Explain how method works. (0.5)
Curve Fitting	Direct polynomial fit	High order fit is useless.
	<i>Lagrange (0.1)</i>	<i>Fix by segmenting. (0.5)</i>
	Segmented polynomial fit	Lagrange reduces roundoff.
	Regression	Distinguish exact fit vs. best fit.
	<i>multiple order (0.5)</i>	Distinguish interpolate vs. extrapolate.
	Spline	Spline matches derivatives on boundary.
<i>multiple order (0.5)</i>	Spline non-unique end conditions.	
Numerical Differentiation	Finite differences	Equally spaced vs non-equally spaced.
	<i>Multiple order (0.5)</i>	Equally spaced, Taylor series (0.5).
	<i>Multiple directions (0.5)</i>	Central difference is better.
	Polynomial fit	Unequally spaced, polynomial (0.5)
		Don’t differentiate data with noise.
Integration of Functions	Simpson’s Rule	Simpson requires equally spaced data.
	<i>1/3 and 1/8 (0.5)</i>	Distinguish accuracy among techniques.
	Trapezoidal	Handling singular integrands.
	Rectangular	
	Quadrature Tech.	
Integration of ODEs	Euler	Distinguish accuracy among techniques.
	Runge-Kutta	Relative speed.
	<i>Multiple order (0.5)</i>	Self-starting vs. non-self-starting
	Finite differences	Initial value vs. boundary value tech.
	Predictor-corrector	Amenable to variable step size.
Shooting	Discontinuities.	

Major Topics	Techniques	Defining Features
Optimization	Gradient descent	Convergence & divergence.
		Multiple optima, initial conditions.
Eigenvalues & Eigenvectors	Jacobi Method	
	Given's Method	
	Householder's Method	
Statistics	Distributions	
	Probability determination	
Fourier Transform	FFT	

For Measure #4, the following connections between topics were awarded credit. Each connection received the same amount of credit.

- *Integration of ODEs* uses *finite differences* and requires solving *systems of linear algebraic equations*.
- Newton-Raphson and Secant methods for *root finding* uses derivatives and *numerical derivatives* of the function.
- *Root finding* uses *curve fitting*.
- *Regression* requires solving *systems of linear algebraic equations*.
- *Direct polynomial fit* requires solving *systems of linear algebraic equations*.
- *Polynomial fit* is used to calculate *numerical derivatives*.
- *Optimization* uses *numerical derivatives*.
- *Optimization* uses *root finding*.
- *Function integration* uses *curve fitting*.