Imagineering Inauthentic Legitimate Peripheral Participation: An Instructional Design Approach for Motivating Computing Education

Mark Guzdial and Allison Elliott Tew
College of Computing / GVU Center
Georgia Institute of Technology
801 Atlantic Drive
Atlanta, Georgia
{mark.guzdial, allison.tew}@cc.gatech.edu

ABSTRACT

When computer science educators are asked to teach non-CS majors, we are often placed in the position of teaching in alignment with a community of practice that does not, or does not yet, exist. In the terms of Lave and Wenger [18], we are not supporting students in legitimate peripheral participation. In that sense, our teaching is inauthentic. We use the example of two classes at Georgia Tech that seem successful by several measures, yet suffer this inauthenticity. We propose that a useful tool for understanding how these classes work is the Disney Corporation's Imagineering—their process of story-telling in three-dimensions as used in their theme parks. However, we find that what students actually learn is not necessarily the story that we are telling them.

Categories and Subject Descriptors

K.4 [**Computers and Education**]: Computer and Information Sciences Education; H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems

General Terms

Experimentation, Design

Keywords

Course design, CS1/2, programming, non-majors

1. LEGITIMATE PERIPHERAL PARTICIPATION

In their book *Situated Learning: Legitimate Peripheral Participation*, Lave and Wenger [18] present a general theory of situated learning. They argue that understanding the social context is critical to understanding learning. They support their argument by focusing on traditional apprenticeship-based learning. From their analysis, they conclude that successful apprenticeship learning occurs through a process of *legitimate peripheral participation* (LPP) in a *community of practice*. In this context, learning means becoming central to the community of practice. This is best supported when new members establish a peripheral, yet legitimate, relationship with that community of practice. Because of their legitimacy, new members are able to grow to become central to the community of practice. While Lave and Wenger limit their examples to apprenticeship learning, they claim that understanding the social context is critical to supporting all learning.

Lave and Wenger draw on several communities of practice to exemplify LPP. The canonical example is the East African tailors that have been used in other studies of apprenticeship [3]. Students begin their apprenticeship by running errands for the master tailors–errands that are *legitimate* because they are authentically needed, but are peripheral to the central practice of being a tailor. However, by running these errands, the apprentices become familiar with the business, language, and practice of tailoring. Later, students will be allowed to cut out single pieces of an outfit, where the design and marking has already been done by the master tailor. Still later, they are allowed to cut out multiple pieces, and at a later stage, assemble pieces. Students in this way progress from peripheral, but legitimate, participation toward full, central participation in the community. Lave and Wenger show how this same model can be used to understand the learning in authentic contexts such as midwives, butchers, and even Alcoholics Anonymous.

This theory has had a dramatic impact in the learning sciences community and is used to help understand many settings (e.g., [16] and [23]). It is generally accepted as helping to explain how learning occurs in authentic practice situations. Lave and Wenger, however, carefully avoid applying their theory to explain traditional schooling, but they do claim that LPP is a *general* learning theory and widely applicable.

In this paper, we explore the application of LPP to schooling, but in a particular *inauthentic* situation: Teaching computer science to non-CS majors. There is no authentic community of practice for most non-CS majors that an introductory course can inform, i.e., there is not a professional community of practice programmers (or even computer science informed professionals) in most liberal arts, management/business, or architecture careers. There certainly are an increasing number of professionals who program. Estimates for the number of potential end-user programmers in the United States range from 55 million [2, 1] to over 90 million by 2012 [22][1]. In

[1]Note however that there is good evidence that few of these end-user programmers understand computer science in a way that we

ICER '06 September 9-10, 2006 Canterbury, UK

our experience, however, most non-CS majors do not see computer science as critical to their professional practice.

Given that we do have non-CS majors in our classes (and may wish more of them, given the drop in enrollments in CS in the Western world [26, 27]), how do we engage and motivate students to learn computer science? Lave and Wenger offer us a theory of what makes instruction engaging and motivating in authentic, professional practices. How can we use the theory of legitimate peripheral participation to design instruction that motivates in inauthentic situations?

In the next section, we will define our terms such that we describe *instruction* which is aligned with a student-perceived community of practice as *authentic instruction*. In Section 3 we describe a successful curricular design sequence—non-CS majors learn computer science concepts, find it motivating, and perhaps even authentic. We then consider in Section 4 how that worked. We designed the course using best practices in learning sciences and participatory design [11], but those practices do not directly address the issue of helping students perceive a community of practice of professionals in their field who use computer science. We instead find that *Disney Imagineering*, the art and practice of creating Disney theme parks, offers insight into our curricular design. Imagineering is about telling a story in three-dimensions over a period of time, and we see natural parallels in the *storytelling* task for creating a successful computer science course for non-CS majors. We conclude with evidence that student learning need not be directly aligned with the story we were telling them.

2. ALIGNED AND AUTHENTIC INSTRUCTION

Lave and Wenger do not explain how their theory of legitimate peripheral participation (LPP) relates to traditional schooling. They emphasize the learning that occurs in everyday social settings. They see school as being too separate from the social practices that give meaning to education [18].

Other theorists who build upon LPP have tried to create a set of terms that explain when schooling is more (or less) a form of LPP. For example, Joseph and Nacu [14] talk about instruction as being "*aligned*" if students can perceive that the school activity leads toward a community of practice that the students value. If students are required to take a course whose goals and purposes cannot be understood by the students as relating to a goal (e.g., joining a community of practice) that the students do not value, then students will not be motivated to learn.

We find Joseph and Nacu's notion of alignment to be similar to Shaffer and Resnick's definition of "*authenticity*" [24]. They talk about four different kinds of authenticity in education:

- Activities aligned with the world outside of school, i.e., do students perceive that the learning activities are aligned with an external community of practice?
- Topics aligned with what learners want to know, i.e., do the students perceive the value of the content being taught?
- Assessment aligned with instruction, i.e., does the assessment measure what was taught?
- Methods of inquiry aligned with the discipline, i.e., are the students learning to think in the modes of a particular community of practice?

might recognize[21].

In general, both alignment and authenticity seem to be describing a similar quality of effective schooling, that is, that students perceive it to valuable because of its connection to an external, authentic community of practice. We make an assumption at this point in our argument. If Lave and Wenger are correct, and LPP is a general theory explaining all forms of learning, then *authenticity is a necessary quality of effective schooling.* If students do not perceive the authenticity of the schooling, then they will not learn. If students learn, then they do perceive that the learning is authentic.

The students may perceive a different alignment than the teacher anticipated or intended. They may be motivated to learn course material as part of a credentialing process necessary to join a community of practice. But however the authenticity is perceived, learning does not take place without it.

3. MEDIA COMPUTATION: SUCCESSFUL BUT INAUTHENTIC

We claim that the non-CS major students do learn computer science in the Georgia Tech courses, CS1315 *Introduction to Media Computation* and CS1316 *Representation of Structure and Behavior*, our two *Media Computation* courses, because the students perceive the courses to be aligned. But they are not—they are inauthentic. The question is how we encouraged the students to believe in an authenticity that was not there. Before we address that issue, we provide evidence to support the claim that students are learning in these courses.

In Fall 1999, Georgia Tech began a new requirement that all students in all majors must take a course in computing. For the first few years, only one course was provided to meet this requirement, and the average withdrawal-and-failure (WDF) rate during this time was 28.2%. In Spring 2003, we introduced two specialized versions of the introductory course—one for Engineers in Matlab, and one for the Architecture, Management, and Liberal Arts majors focused on using programming to manipulate media [5, 8, 20]. Both courses have improved success rates, with the media course losing only 10-15% (WDF rate) of the students each term [6].

The goal of the Media Computation approach is to teach computing in a context that students find relevant—that students see it as aligned with a community of practice that they value. Students in Architecture, Management, and Liberal Arts majors typically use the computer as a tool of communication, rather than a tool for calculation [5]. Thus, we teach the same CS content as in a normal CS1 and CS2, but using media as examples.

Students in the first course use Python to manipulate and create pictures, sounds, text, and video and animations (frames). They learn iteration by computing grayscale images, to concatenate arrays by splicing sounds, and to compose strings by generating HTML pages from data [9]. When we teach the course, we tell the students about the value of being able to manipulate media apart from any tool. We point out to them the many media companies that use Python, including Pixar and Industrial Light & Magic. We try to connect the course to the community of practice of professional media developers.

The second course was created for students interested in going on to further computer science courses. The second course focuses on data structures, rudimentary object-oriented programming in Java, and simple design notation (UML class diagrams), though still in a media context. Students use linked lists to construct animations and trees to implement scene graphs. Students implement continuous and discrete event simulations in order to create animations, and in so doing, create authentic contexts for learning about stacks and queues.

The driving question for the second course is, "How did the wildebeests charge over the ridge in *The Lion King*?" We start the course showing the scene from Disney's *The Lion King* where wildebeests charge and stampede the main character's father. We then show a scene of villagers in a square from Disney's *The Hunchback of Notre Dame*. These two scenes were the first ones in which Disney modeled the characters on the computer, rather than draw them by hand. The villagers and wildebeests were then simulated in order to generate the animation. The entire second course is related to the goal of generating villagers and wildebeests—a wildebeest even graces the cover of the course notes. Linked lists and trees are related to modeling the characters, and stacks and queues are related to the simulations. The second course is clearly connected to the community of practice of professional computer animators.

3.1 It's Successful

We began the first course in Spring 2003, and the second in Spring 2005. Each course has been offered every academic term since then. Some of our findings from our studies of the effectiveness of the approach are:

- Retention has improved dramatically, from a 72% success rate (earning A, B, or C) to 85-90% success rate in both courses, with similar results in other adopting schools [25]. The introductory course is about 300 students/semester, and has an average 51% female population. The second course was 75% female in its first offering.

  These retention statistics actually underestimate the impact of the course on non-CS major students. Within non-technical fields, the impact of the Media Computation course has been much higher—see Table 1[2].

- Women in the media computation course report in interviews finding the course more relevant than do women in the traditional first course [20].

- Women report finding the course to be creative with a rich social context supported by an on-line environment for sharing media [5, 11]. Quoting from one female liberal arts student in an interview:

  > "I just wish I had more time to play around with that and make neat effects. But JES [IDE for class] will be on my computer forever, so that's the nice thing about this class is that you could go as deep into the homework as you wanted. So, I'd turn it in and then me and my roommate would do more after to see what we could do with it."

A year after the first offering of the course, we conducted an email survey of the students that had taken the course. Over a quarter (27%) of the respondents had manipulated new media since leaving the class, and 19% of the respondents had actually written programs since class had ended, mostly to manipulate media [11]. In particular, students told us how much the class impacted how they interacted with computation.

> "Definitely makes me think of what is going on behind the scenes of such programs like Photoshop and Illustrator."

[2]Biology appears in this table because Biology gave their students the option of any CS1 at Georgia Tech, and the majority chose Media Computation.

Table 1: Average non-CS majors success rates, in traditional CS1 (Fall '99 to Spring '03), and since, in *Media Computation* (Spring '03 to Fall '05)

Major	*Traditional CS1*	*Media Computation*
Architecture	46.7%	85.7%
Biology	64.4%	90.4%
Economics	54.5%	92.0%
History	46.5%	67.6%
Management	48.5%	87.8%
Public Policy	47.9%	85.4%

Table 2: Success rates of *Media Computation* students in the second CS course for majors compared to all students

Semester	*Media Computation Students*	*All Students*
Summer 2005	100% ($n = 2$)	87% ($n = 102$)
Fall 2005	57% ($n = 7$)	61% ($n = 328$)

> "I understand technological concepts more easily now; I am more willing and able to experience new things with computers now"

> "I have learned more about the big picture behind computer science and programming. This has helped me to figure out how to use programs that I've never used before, troubleshoot problems on my own computer, use programs that I was already familiar with in a more sophisticated way, and given me more confidence to try to problem solve, explore, and fix my computer."

Since some students self-identify at the time of a major change, we do know that we have students changing their majors into computer science, and our new *BS in Computational Media* major. However federal privacy regulations prevent us from tracking specific students without explicit consent, so we are unable to identify how many students have been attracted to computing majors through this introductory course sequence.

So far, nine students have gone from CS1316 into our traditional second course for CS majors. (Another eight are enrolled this semester.) Table 2 summarizes the success rates of the students from the Media Computation data structures course compared to all students in this second course. While the numbers are low, the trend is encouraging. Overall, the second course for majors is a difficult course with a low success rate, but the Media Computation students are comparable to the other students—they are not less well-prepared and they succeed at about the same rate.

We believe that the above suggests that students *are* learning computer science in these courses. First, students are clearly *succeeding* at these courses (e.g., passing) where many had not previously. Second, they are reporting continued use of computing and an effect on their daily lives (a key point that we will return to later). Third, they are successfully blending in with CS majors in more advanced courses. Some are even becoming computing majors, which suggests a strong sense of belonging to a community of practice that they may not have considered before the course.

3.2 But It's Inauthentic

But it's a lie—a story we tell them. Nobody processes media at the level of pixels and samples in Python. In fact, very few people write programs to manipulate media at all! Virtually all professionals who were architecture, management, and liberal arts majors use commercial software like Photoshop for manipulating media. There is not a perceivable community of practice for these students to belong to. People at Pixar and Industrial Light and Magic who create animations do not use Java. The students are using inauthentic tools for inauthentic tasks. What we have done is to construct an *illegitimate* peripheral participation—seemingly, we have convinced students that this is authentic, that they are moving toward the center of a community of practice that they value.

The question we are addressing in this paper is not the ethics of what we are doing. One might similarly question the ethics, from an LPP perspective, of requiring any course of study of non-majors where they cannot perceive the relationship to a community of practice that the students value. We think about our task as creating a community of practice that does not yet exist. Professionals from architecture, management, and liberal arts may not, as a whole, use computing in a powerful way yet—but they might. By teaching computing to these students, we create the opportunity for such a community later.

How did we convince the students of the authenticity of what we were teaching them? What we were doing was not traditional instructional design, because it was not about transforming mental states but motivating that transformation [7]. One might consider what we were doing as marketing. While marketing is about finding out what the consumer needs and meeting those needs, most of our issues are not marketing issues—we're not concerned about pricing, distribution, or promotion [15]. What we are doing is storytelling [4, 19], but storytelling that takes place in real space over a the length of a semester, as opposed to storytelling in the more traditional media forms of books, movies, or television.

4. APPLYING IMAGINEERING TO EXPLAIN MEDIA COMPUTATION

We believe the storytelling method that best explains what worked in the Media Computation classes is *Disney's Imagineering* [12, 13]. When Walt Disney set out to create Disneyland (and later, Disney World), he recruited engineers and animators who then learned to tell stories in the three-dimensional space of a theme park. They created rides and spaces to walk around in (like Main Street, Frontierland, and Tomorrowland) that tell stories across these spaces.

We have identified six processes that Imagineers use in telling their stories in theme parks which we believe are in common with what we are doing in the Media Computation classes.

1. Start from the story.
2. Start from where the expectations are.
3. Pay attention to details.
4. Where necessary, change reality to support the story.
5. Pay attention to transitions.
6. Make the cast part of the story.

Creating a course has much in common with designing a theme park. The students' experience occurs in three-dimensions in multiple spaces: from the lecture hall, to the laboratory, to the dorm room. The experience occurs over time, not the hours or days of a theme park visit, but the 15 weeks of a semester. We will now explain how each of these processes appears in the Media Computation courses.

4.1 Start from the story

Everything in the Disney theme parks starts from a story [12]. Before the Big Thunder Railroad roller coaster was created, a story was generated that explained how this ride was an accidental trip through an abandoned mine. That story defined the story elements (e.g., signs and animated characters, even plants and scenery) that went into the roller coaster. The futuristic Tomorrowland was always a problem for Disneyland and Disney World because tomorrow kept arriving. In the latest revisions to these theme parks, the Imagineers made an explicit decision to make Tomorrowland retro-futuristic—it is based on a view of the future defined in the 1950's and 1960's [12, 17]. Coming up with a new story literally defined and limited a large revision to the theme park.

Even the definition of vendor areas starts from a story. The Emporium gift shop in the Main Street area of Disney World was designed based on a story [13]. The Emporium spans several shop areas on Main Street. The Imagineers developed a story about a proprietor who's good fortune led to expansion through adjoining shops that he was able to purchase over a period from the late 1800's through the early 1900's. Each of the shops in the Emporium is decorated to depict the decade in which the shop was supposedly purchased. Even the light fixtures are meant to depict these decades, from fixtures meant to look like gas lighting, to fixtures meant to depict hybrid electric-gas lighting, to early 20th century light fixtures.

In the introductory course, we tell a consistent story that was developed during the initial definition of the course [8]:

- All media are going digital.
- Digital media are manipulated in software.
- Knowing how to program, then, is an advantage in a profession that manipulates media.

That story is often repeated in the course and has driven the design from the very beginning. We very explicitly connect the course to the community of practice of digital media professionals.

As mentioned, the second course tells a consistent story from the very first day of the course. It's all about the wildebeests and the villagers. We repeat that connection throughout the course–it's all about the community of practice of computer animators.

4.2 Start from where the expectations are

In the story of the creation of the Boardwalk area in Disney World (Figure 1), Jeff Kurtti writes [17], "Just as Main Street, U.S.A. in the Magic Kingdom and Hollywood Boulevard at Disney-MGM Studios are not meant to represent factual history, but to evoke a collective cultural memory, the flavor of the 1920s mid-Atlantic coast is apparent at Disney's BoardWalk." Imagineers are not aiming for authenticity in what they do. They are aiming for *perceived* authenticity—meeting people where they are.

This process is really about peripheral participation. The places feel familiar because they match past experience. Meeting people's expectations is about saying to them, "You are already part of this place, or this community of practice. You are in the periphery. Come visit and explore." Perhaps by doing so they will even move toward more central participation in the community.

We tell students in both Media Computation classes that they have been peripherally participating in the media manipulation community of practice already. We ask them in class about the media that they already have on their computers. We ask them about what

Figure 1: Disney BoardWalk at night, featuring a dance hall next to brightly lit shops, because that's what one would expect on a BoardWalk

they do with Photoshop and iPhoto, then relate the algorithms being learned to those applications. We encourage students to use their own media in their homework assignments. We connect the course forward to the communities of practice about which we are storytelling. When we introduce the chromakey algorithm in the first course, we bring in the movie *The Making of The Matrix* and point out where chromakey is used in special effects there. In the second course, we regularly review scenes of wildebeests and villagers, and talk about scenes from *The Incredibles* and other animated movies.

4.3 Pay attention to details

The details that Imagineers manipulate in construction of the theme parks may oftentimes seem too subtle to make much difference. But they believe that paying attention to such details means that some are going to be noticed, and that that will enhance the experience tremendously [12]. For example, around the Aladdin's Magic Carpet ride at Disney World is a bazaar filled with vendor's carts selling jewelry and other knick-knacks meant to evoke the time and place of Aladdin—a mystical, medieval Arabic culture. The paving stones near the ride actually have cemented into them some of the same jewelry as can be found on the vendor's carts–suggesting the story that the jewelry fell into the dirt thousands of years ago, and was ground into the dirt [13].

We have paid similar attention to the details in the design of these two courses. For the introductory course, we wrote the lectures and slides, which match the book that we wrote [9], which tie tightly to the assignments (e.g., students are asked to make an image collage after learning image filters and manipulations), and examples of those assignments from past semesters can be found in the on-line Galleries [20]. The examples from the book appear in the lectures, and the source media are on the CD that the students' receive so that they can use the media in their own work. The programming environment, JES, was created explicitly for the course [8], and it includes supports for media manipulation (e.g., the ability to investigate the RGB values of individual pixels). Screenshots that show JES being used for course examples appear in the book.

The story is told consistently and becomes self-supporting pieces of evidence. One might imagine a student thinking, "Why, *of course* people manipulate media with people! Look at all the great things in the Galleries—those were all done with Python. Look at these examples in the book and lecture notes—these look the kinds of things professional media manipulators do. JES works fine for media, and I can do professional-looking things with it."

4.4 Where necessary, change reality

The first author's interest in how Imagineers did their work started on a family trip to Disney World where we stayed at the Wilderness Campground. The trip to the Magic Kingdom from the Wilderness Campground is by boat. Once the boat leaves the stop at the Wilderness Resort and makes the turn toward the Magic Kingdom, Cinderella's Castle, the representative icon of the Magic Kingdom story, is visible in the distance (Figure 2). It is continually visible, and grows larger while moving toward the Magic Kingdom—a constant reminder of the story that we were entering. But in passing the Contemporary Resort on the right, we noticed cars in front of the resort. How did those cars get there? Then we noticed the tunnel *under* the waterway. In most places, if a road has to cross a waterway, the road goes over the waterway. But at Disney World, that would break the visual connection to the story, so the waterway takes precedence.

Figure 2: Main Street in Disney World's Magic Kingdom, with Cinderella's Castle in the background

In a similar way, Disney Imagineers fiddle with basic assumptions and standard practices in order to convey the story and feel for which they aim. Another example is the manipulation of proportions along Main Street (Figure 2). It is important for the icon of the Magic Kingdom, Cinderella's Castle, to be always visible while entering the Magic Kingdom. But it is also important for Main Street, the entranceway, to feel warm and inviting. But expectations are that buildings on 20th century American Main Street's were three stories tall. A three story building would block the castle and feel intimidating. Imagineers use perspective proportions to shrink the second and third stories of the Main Street buildings, so that the buildings *look* like three stories, but are actually smaller (Figure 3) [13].

Of course, manipulating reality in software is much easier than in the physical world. For us, creating a reality that supported our story did not involve building specially-sized buildings or underwater tunnels.

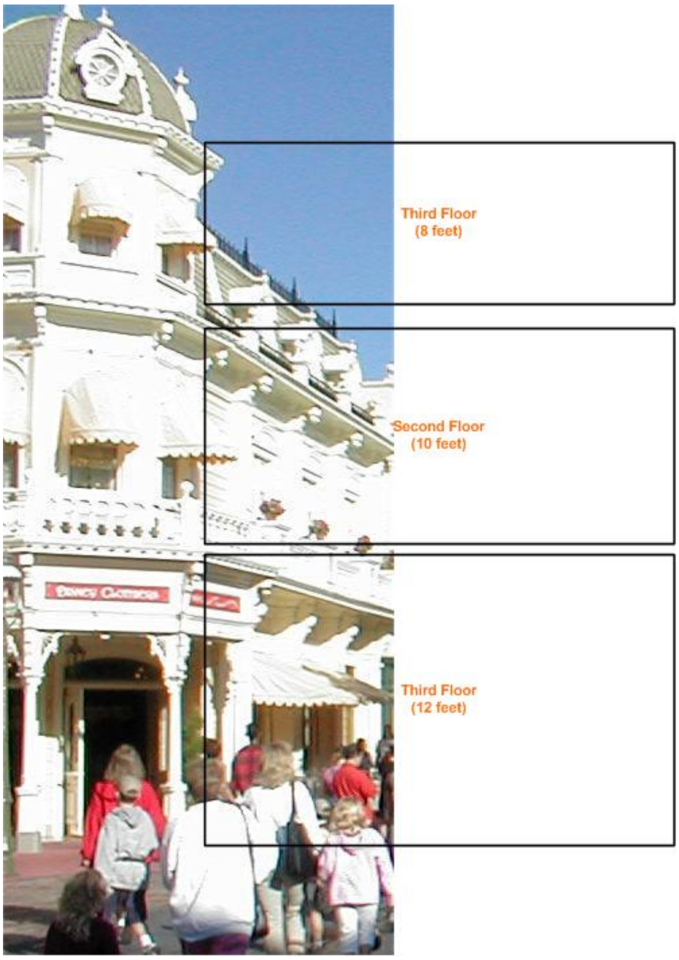

Figure 3: Measuring the Floors on Main Street buildings

The reality is that basic Python does not support media manipulations. But to convey the story of the media computation classes, we wanted to make it effortless and integrated. We created libraries for media manipulation and the programming environment with media supports built-in (e.g., ability to zoom pictures and check RGB values of pixels and the ability to look at graphs of sounds and individual sample values) [11]. Behind the scenes, invisible to the user, the programming environment `imports` the media libraries, so that the student never sees that media manipulation is something extra.

The reality is that Java media support is complicated, with classes like `MediaConsumer` and `BufferedImage`. We created new classes with names like `Picture` and `Pixel` that hide the multiple classes needed to manipulate media in Java [10]. We direct students on the first day to add these classes to their `classpath`, and from then on, it is invisible how real media manipulation in Java works.

We recognize that students do have to leave the class one day and work in the real world. To that end, the Python book does explain how to import the media libraries into traditional Python [9]. The Java book does use traditional Java libraries for 2-D graphics, so that students know how to use the actual Java API [10]. But as much as possible, we try to keep the story consistent during the class. From their perspective, it seems quite reasonable to believe that the community of practice of professional media developers would work just as they do.

4.5 Pay attention to transitions

One of the more popular locations in Disney's Fantasyland is the Enchanted Tiki Room where audio-animatronic (robot) birds sing. The setting for the Enchanted Tiki Room is somewhere in Polynesia. The roof of the Enchanted Tiki Room is capped with four water buffalo. Why water buffalo? Why not some other Asian animal? Because the top of the Enchanted Tiki Room is visible from Frontierland (modeled after visitors' perception of the American wild West), and from that distance, the water buffalo look like longhorn steer.

Imagineers worry about sight lines and about transitions from one land to the other. They want the transition to be smooth, but more importantly, they want you to understand the new story that you are entering. Within a story, everything that you see should be consistent with that story.

We do similar things in the Media Computation classes. As each new topic in the course is introduced, we relate the topic to the story. We do not teach string processing in the first course—we start teaching how to generate HTML pages which can hold our media. We do not teach linked lists in the second course—we teach how to dynamically and creatively insert and remove media elements (music, images, etc.) without having to move everything else in the array. We are continually relating the topics of the course back to our story about a community of practice of professional media developers.

4.6 Make the cast part of the story

Disney World has the world's largest working costuming shop. Everyone at Disney World is "themed"—dressed to match the story of where they work. This is not just for the characters. For example, the costumes of the salespeople in the Emporium match the decade in which their shop was purchased by the mythical proprietor of the Emporium.

In the Media Computation classes, we draw the students, and the teaching assistants, too, into the story. After assignments are turned in, we might visit the on-line Gallery page live in lecture, find a particularly good one, then ask, "This is a great collage on the Gallery! Who did this? You did? How did you do it? How did you make this effect here?" The student starts explaining what she did, without realizing that she has now bought into the model of talking about media manipulation code as a professional in her community of practice. Teaching assistants will regularly solve the assignment for themselves and post it–often creating some of the best work, but also inadvertently buying into the story of creating media with code. Thus, the people in class with the students are part of this community of practice in which they are all participating.

5. CONCLUSIONS: A WILLING SUSPENSION OF DISBELIEF

When people leave Disney World, they do not leave believing that mice are six feet tall and can talk. They might leave with some pleasant memories of good times, but in general, Imagineers are not aiming for learning. We are–we want the students to take something away from the story.

There is good evidence that we are having an impact, but the students are not believing the story. There may be a willing suspension of disbelief during the class, but after the class, they know what it was all about. They did not believe the story that we were telling them about a community of practice of media developers who use

Python and Java for manipulating digital media. However, they did learn, and there's evidence that they perceived a more powerful community of practice that did drive their learning.

Consider the follow-up survey comments described earlier (from [11]). Note that the students are not saying that they learned a lot about computation for their future profession. They are telling us that they learned a lot about computation that influences their daily lives.

In our comparative study between Georgia Tech and Gainesville College students using Media Computation [25], we found that, in general, students did not find the homework "relevant." Only 39% of the students at Georgia Tech and 31% of the students at Gainesville found the homework relevant. Further, a majority did not find that the skills learned in the class would be useful in their future careers—45% of the students at Georgia Tech and 37.5% of the students at Gainesville thought that the course skills and concepts would be useful in their future career. However, the majority of Georgia Tech students (59.9%) and Gainesville students (56.2%) said that they thought that the "skills would be useful later in life."

We believe that the storytelling in the Media Computation classes made the course successful and palatable, but we did not convince anyone of the story. Rather, the story helped in motivating the real learning—computing skills for life. In a real sense, the class became a "computing appreciation" class, but without the negative connotations. As one student put it in the follow-up survey [11]:

> "Other than making me a little more aware about what I can make the computer do, it hasn't changed the way I particular interact with technology. Yet I am uninterested in this field. However, I now have a MUCH better understanding of the people who are interested in this field, how they view things, and how to interact with them more easily. For this, I appreciate the CS class greatly."

Thus, the students seem to perceive that the courses were helping them to become more central in a community of practice that related more to daily life. Perhaps they saw themselves as members of a community of practice of informed professionals, or perhaps of media and technology literate citizenry. What actual community of practice the students perceived is worth further investigation. However, we contend that the storytelling techniques that Imagineering offers helps to convey a sense of a community of practice that helped to sustain student motivation through the courses.

6. ACKNOWLEDGMENTS

This research is supported in part by a grant from the National Science Foundation CISE EIA program, and by the College of Computing and GVU Center.

7. REFERENCES

[1] Boehm, B. W., Clark, Horowitz, Brown, Reifer, Chulani, Madachy, R., and Steece, B. *Software Cost Estimation with Cocomo II with Cdrom*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.

[2] Boehm, B. W., Clark, B., Horowitz, E., Westland, J. C., Madachy, R. J., and Selby, R. W. Cost models for future software life cycle processes: COCOMO 2.0. *Annals of Software Engineering 1* (1995), 57–94.

[3] Collins, A., Brown, J. S., and Newman, S. E. Cognitive apprenticeship: Teaching the craft of reading, writing, and mathematics. In *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser*, L. B. Resnick, Ed. Lawrence Erlbaum and Associates, Hillsdale, NJ, 1989, pp. 453–494.

[4] Davis, D. *Telling Your Own Stories*. August House, 1993.

[5] Forte, A., and Guzdial, M. Computers for communication, not calculation: Media as a motivation and context for learning. In *Hawaii International Conference on System Sciences* (2004).

[6] Forte, A., and Guzdial, M. Motivation and non-majors in computer science: Identifying discrete audiences for introductory courses. *IEEE Transactions on Education 48*, 2 (2005), 248–253.

[7] Gagne, R. M., Wager, W. W., Golas, K., and Keller, J. M. *Principles of Instructional Design*. Wadsworth Publishing, Belmont, CA, 2004.

[8] Guzdial, M. A media computation course for non-majors. In *ITiCSE '03: Proceedings of the 8th annual conference on Innovation and technology in computer science education* (New York, NY, USA, 2003), ACM Press, pp. 104–108.

[9] Guzdial, M. *Introduction to Computing and Programming in Python: A Multimedia Approach*. Prentice-Hall, Upper Saddle River, NJ, 2004.

[10] Guzdial, M., and Ericson, B. *Introduction to Computing and Programming in Java: A Multimedia Approach*. Prentice-Hall, 2005.

[11] Guzdial, M., and Forte, A. Design process for a non-majors computing course. In *SIGCSE '05: Proceedings of the 36th SIGCSE technical symposium on Computer science education* (New York, NY, USA, 2005), ACM Press, pp. 361–365.

[12] Imagineers. *Walt Disney Imagineering: A Behind the Dreams Look at Making the Magic Real*. Disney Editions, Orlando, FL, 1998.

[13] Imagineers, T. D. *The Imagineering Field Guide to Magic Kingdom at Walt Disney World*. Disney Editions, Orlando, FL, 2005.

[14] Joseph, D., and Nacu, D. C. Designing interesting learning environments when the medium isn't enough. *Convergence 9*, 2 (2003), 84–115.

[15] Kamins, M. A. *Sales and Marketing: The New York Times MBA Series*. New York Times, New York, NY, 2003.

[16] Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., Puntembakar, S., and Ryan, M. Problem-based learning meets case-based reasoning in the middle school science classroom: Putting learning by design (tm) into practice. *The Journal of the Learning Sciences 12*, 1 (2003).

[17] Kurtti, J. *Since the World Began: Walt Disney World, The First 25 Years*. Disney Editions, Orlando, FL, 1996.

[18] Lave, J., and Wenger, E. *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press, Cambridge, UK, 1991.

[19] Mooney, B. *The Storyteller's Guide*. August House, 1996.

[20] Rich, L., Perry, H., and Guzdial, M. A cs1 course designed to address interests of women. In *Proceedings of the ACM SIGCSE Conference* (2004), pp. 190–194.

[21] Rosson, M. B., Ballin, J., and Rode, J. Who, what, and how: A survey of informal and professional web developers. In *Visual Languages and Human-Centric Computing, 2005 IEEE Symposium on* (2005), pp. 199–206.

[22] Scaffidi, C., Shaw, M., and Myers, B. Estimating the numbers of end users and end user programmers. In *Visual*

Languages and Human-Centric Computing, 2005 IEEE Symposium on (2005), pp. 207–214.

[23] Scardamalia, M., Bereiter, C., and Lamon, M. *The CSILE Project: Trying to bring the classroom into World 3*. MIT Press, Cambridge, Mass., 1994, pp. 201–228.

[24] Shaffer, D. W., and Resnick, M. "thick" authenticity: New media and authentic learning. *Journal of Interactive Learning Research 10*, 2 (1999), 195–215.

[25] Tew, A. E., Fowler, C., and Guzdial, M. Tracking an innovation in introductory cs education from a research university to a two-year college. In *SIGCSE '05: Proceedings of the 36th SIGCSE technical symposium on Computer science education* (New York, NY, USA, 2005), ACM Press, pp. 416–420.

[26] Vegso, J. Interest in cs as a major drops among incoming freshmen. *Computing Research News 17*, 3 (2005).

[27] Vegso, J. Bureau of labor statistics projects it workforce to add a million new jobs between 2004 and 2014. *CRA Bulletin* (2006).