Early Insights into Teaching Computer Science with Story to Middle School Girls

Gail Carmichael, Elizabeth Stobert, Sarah Lacelle, David Mould, Robert Biddle
Carleton University, Ottawa, Canada

ABSTRACT

Stories have been the vehicle of human knowledge for many generations, and some CS educators have been turning to them to help improve their pedagogy. We wanted to learn more about the experience of using story to teach introductory computer science topics, so we conducted a qualitative observational study with middle school girls. We taught two CS Unplugged activities -- Image Representation and Finite State Automata -- with a story and a non-story condition. We found that while story was engaging and resulted in better understanding than an abstract activity, it did not necessarily have any advantages beyond having a light context. Our insights into the student and instructor experience will help guide future study on story.

INTRODUCTION AND BACKGROUND

Much knowledge has been passed through the generations in the form of stories. In a SIGCSE Bulletin invited editorial, Papadimitriou (2003) explained his belief that story and context can make computer science more interesting and easier to learn and remember. He envisions placing ideas into historical contexts, illustrating concepts through narrative, and embedding lessons into stories. We are interested in how we can use Papadimitriou's third approach to teach introductory computer science skills to middle school girls. Insight into both the students' and instructors' experience of using story may help educators integrate story into their own lessons as one way to help improve computer science education and increase participation of women in the field.

There are several existing examples of educators and authors using a combination of Papadimitriou's approaches to teach computer science to a variety of audiences. Some researchers employ narrative as a connective thread between concepts in a course. For example, Christensen (2009) reports on the use of story in a software engineering course. Each class introduced a new chapter in the story of a company asked to develop a parking lot pay station, ensuring that the problem was studied before the solution is studied. This approach succeeded in demonstrating how the techniques being taught could effectively be used together.

Yarosh and Guzdial (2007) also applied context as a connective thread in their second level data structures/media computation class. Their central question of how animators created the wildebeest stampede scene in The Lion King interested most students and motivated learning, even though few found it particularly relevant.

Story can independently motivate individual topics as well. Rao (2006) describes how stories and puzzles were first used to break up long video-linked lectures and later integrated into traditional classrooms. The stories helped motivate the concepts at a higher level of abstraction and increased the students' interest. Better grades and a higher quality of work in student projects resulted.

Stories have been shown to improve memory for a variety of subjects. For instance, in a psychology course, students who read narrative rather than expository text performed better on quizzes and later recalled information more easily (Janit, Hammock, & Richardson, 2011). In one computer science class, stories acted as a useful mnemonic device that most students found useful in understanding and remembering material (Pollard & Duvall, 2006).

Authors are also starting to use Papadimitriou's third approach by embedding computer science content into fictional stories. For example, Lauren Ipsum (Bueno, 2011) is a short novel written for children. It follows the journey of Lauren as she tries to find her way home from Userland. To succeed, she must learn about a wide variety of computer science topics, from efficient algorithm design to flipping fair coins. Computational Fairy Tales (Kubica, 2012) provides another example of this third approach. It is intended to motivate computer science topics through a series of short, interconnected stories involving fairy-tale characters like dragons and wizards. It is also intended for a pre-college audience, and has been useful in classrooms as a way to motivate topics before delving into the details (Kubica, Who is the target audience?, 2012).

While many of these approaches may be useful in recruiting middle school girls to the field of computer science, there have been efforts to use story specifically to accomplish this goal. For example, Storytelling Alice (Kelleher, Pausch, & Kiesler, 2007) motivates middle school girls to learn programming by allowing them to create 3D stories with code. The girls learned programming equally as well with Generic Alice, a version of the program without storytelling, but spent more time working on Storytelling Alice. In our study, we want to determine whether story can have an impact on learning general computer science concepts.

Stories typically involve many elements, such as plot, character, conflict, and timeline, but some of these elements can be integrated into a teaching environment without involving a fully-fledged story. In the work presented here, we adopt Bal's definition of story as a particular presentation of "a series of logically and chronologically related events that are caused or experienced by actors" (1997, p. 5). We defined context as a light fiction that relates educational content to a concrete scenario. While stories have context, not all context is part of a larger story. We were interested in the experience of teaching and learning with Papadimitriou's third approach of embedding content into stories. We conducted a study where we taught new concepts using stories, and observed how this affected the experience of learning and teaching introductory computer science topics as compared to a non-story approach.

STUDY

The goal of our study was to understand how story can be used to help teach computer science topics. To investigate this question, we conducted a week-long observational study with middle-school girls. Our study was designed to examine the experiences of learning and teaching both with and without a story. The study had two conditions: *story* and *non-story*. The story condition integrated complete stories into lessons, and the non-story condition used lessons that included context but not the more immersive elements of story.

We conducted our study as part of a week-long mini-course on computer science and games that was aimed at middle and high school girls (Carmichael, 2008). Each year, the Enrichment Mini-Courses Program (http://www.carleton.ca/emcp) is offered by post-secondary institutions in our area. The program invites students from nearby schools to participate in week long courses on a variety of topics, and the courses are mainly taught by graduate students. Our course was taught by two of the authors of this paper who are PhD students in computer science.

Students from middle and high schools are selected by their teachers to attend the mini-courses, and course assignment is based on a system where students are asked to list their top preferences, but are not guaranteed to get one of those choices. Once assigned, students are not permitted to change courses except in exceptional circumstances. Our course was restricted to girls, on the basis that they were more likely to want to explore computer science topics in an environment that would not be male-dominated. There were 22 girls in our course, and all but one participated in our study. The majority of girls were in grade 8 (aged 12--13), but one was in grade 9, and one in grade 11.

The study was conducted over two lessons that took place on the Wednesday and Thursday of the course. Participants were randomly divided into two groups, and on each day, one group completed the story condition and the other group completed the non-story condition. Each group was exposed to both the story and non-story conditions. The instructors and observers in both conditions remained the same for the two activities. Below, we describe the activities and the stories that accompanied each one. Participants were made aware of the study prior to the beginning of the course, and we requested the informed consent of both the girls and their parents/guardians. The study was approved by our university's ethics committee.

Each activity began with a lesson, and ended with an assessment worksheet. Observers noted the kind of behavior seen in the room, and recorded questions and comments from students. The assessment worksheets were also collected, and the responses were analyzed to give insight into participants' learning and understanding. We also collected observations from the instructors.

We analyzed the data using a qualitative analysis approach. We began with a grounded theory approach using open-axial coding, but did not follow it strictly. Each observation was analyzed and coded according to the kind of learning that we saw happening. The same was done for the responses on the worksheets. After coding all of the responses, we reviewed our codes, and identified several relationships in the data, as well as key quotes and observations that supported our findings.

The Activities and Worksheets

We based our lessons on two Computer Science Unplugged (Bell, Fellows, & Witten) activities: Image Representation and Finite State Automata. We chose to use existing activities rather than design our own because we wanted to ensure the activities themselves did not affect the outcome -- CS Unplugged activities have been used extensively, and we have used them successfully in past versions of the course.

We used largely the same examples and exercises for both the story and non-story conditions. Exceptions included a story-related example in the Image Representation activity, and the substitution of 'H' and 'T' for 'A' and 'B' in the Finite State Automata activity.

At the end of the lessons, students completed evaluative worksheets. We designed these worksheets, and they were the same for both groups. The Image Representation worksheet focused on the two ways of representing black and white images presented in the lesson: writing a one or zero for each black or white pixel in every row, or alternating between the number of black or white pixels that appear together in a row. We asked knowledge and understanding questions centered on the pros and cons of these representations. The Finite State Automata worksheet asked about words (routes on the map) and languages (set of valid routes on the map). It also had some puzzles from the CS Unplugged activity, where the students are asked to find the pattern in the words part of a particular automata's language. For each response on both worksheets, we assigned a score from 0 (no answer) to 4 (correct).

The Stories

In the story condition, one of the instructors outlined a story for each of the topics before class, and then told the story in front of the group. This instructor had some previous writing and performance experience. The following story summaries situate the relevance of certain results and offer specific examples of how computer science content might be embedded into a story.

Image Representation and The Helicopter Man

This story follows the journey of a village girl who is on a quest to find her missing father. After setting out, the girl befriends a boy who informs her of a strange man with a helicopter pack. This man was seen around the time people began to disappear from their villages. Apparently the strange man was known to understand words and pictures, but not numbers. The two part ways.

The girl eventually reaches a fortress where she hears her father's voice. A paper airplane flies out his window; the page contains rows of ones and zeros. The girl tucks it into her pack when the helicopter man suddenly appears, hovering over her. He had been expecting her, and says he will give her a chance to earn her freedom and rescue her father if she can escape his carefully designed maze. The man sees her page of ones and zeros, but thinks nothing of it. The girl must determine that the numbers represent an image of the maze -- a map to help her find her way out.

[An example map is shown in class and the students discuss how a black and white image can be represented with ones and zeros.]

While navigating the maze, the girl finds that the boy she met earlier had also been captured. The two come up with a plan to escape the maze and rescue the girl's father. The girl creates a map to help the boy find her village after their plan is executed. But this map needs to be much larger than the maze map, so they must find a way of writing down fewer numbers onto the page.

[At this point, the students are asked how this compression might be accomplished, and then have an opportunity to practice with an image of a compass rose, representing part of the complete map. After a quick conclusion to the story, the students complete the activity's worksheets.]

Finite State Automata and The Great Island Race

This story takes place on a distant and remote island where a family maintains a large plot of land farmed for income and to feed themselves. Recently, some greedy neighbors decided to take over most of their most fertile land “for the good of the village.”

The teenaged daughter of the family loves to explore the island and has been taught survival skills to keep her safe. One day, she is exploring the far side of the island when a small, rickety boat sails in. A boy holding parchment steps off, lost in his own thoughts. When he fails to notice a wild dog about to attack him, the girl takes action and saves him.

When asked what he is doing on this side of the island, the boy tells the girl that he is creating maps for his father's secret project. He briefly shows her the map as thanks for saving him; she notices islands marked with circles and arrows labeled ‘H’ and ‘T’ in between. [An example map is shown to the participants.]

Days later, an official from a nearby village announces a competition for the surrounding islands. It will involve both speed and luck: contestants will sail swiftly between islands, and toss a coin to learn where to travel next. Everyone will start on Pirate's Island, and whoever reaches Treasure Island first will win.

The daughter decides to enter along with her brothers, hoping that a win could help reclaim the family's land. Remembering the map the boy was making, she tracks her progress through the islands on the day of the contest and has her brothers do the same.

[The story pauses here and the students complete the pirate map activity, labeling their routes through the islands with `H' for heads and `T' for tails.]

At the end of the competition, the winner by a large margin is revealed to be the man who took the family's land. Suspicious, the daughter puts together her and her brothers' maps to see what routes through the islands are even possible (the students do the same at the front of the class). She then asks the officials at each island when the winner had passed through. She realizes that he must have cheated, because the route he took is not a valid route according to their maps.

Thanks to this discovery, the winner is sent to jail, the family's land is returned, and the girl is hired as a map-maker for future competitions, allowing her to explore further.

[After some brief discussion of finite state automata and languages, the students fill in their worksheets.]

RESULTS

From the worksheet results, observations of the students during the lessons, and the instructors' reflections on teaching, several categories of open codes emerged. The key categories of codes are *understanding* (deep to missing), *articulation* (good to poor), *representation* (concrete to abstract), and *engagement* (engaged and interested). Each of these represents a range except for engagement. The numerical scores assigned to worksheet responses (summarized in Figure 1) give further indication of understanding. Details of the coding and worksheet results follow.

Figure 1: Histograms of students' worksheet scores.

The Student Experience

Story

Students in the story condition demonstrated better understanding in the Image Representation worksheet responses than for Finite State Automata based on the use of higher understanding tags and the results in Figure 1. In some cases, students used story-related language when developing their understanding for both topics. For example, when asked what words in a language represent on the pirate's map on the Finite State Automata worksheet (answer: routes through the islands), several students referred to heads and tails. One student wrote:

> "Which way you're going. Heads going one way and tails going another way."

During the evaluative Image Representation activity done before the worksheet, several students demonstrated their understanding using images related to the story. One image contained the word "HELICOPTER" (see Figure 2), and another had the words "THANK You For HELPIN" in reference to the characters working together to escape the maze.

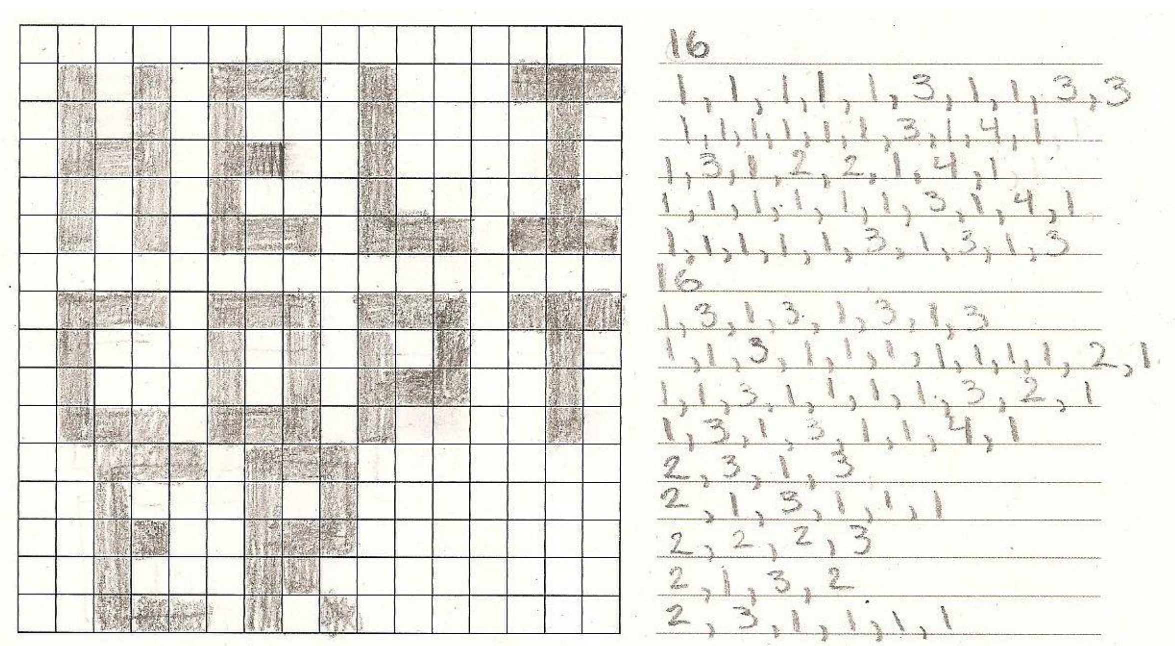

Figure 2: An image representation worksheet result.

Similarly, some of the sudden insights students had during the lessons were also expressed with story-related language. One student was able to determine the meaning of the first image shown during the Image Representation lesson and said, "All the 1's are the bricks, all the 0's are the pathways." In the same activity, when filling in the practice image of a compass rose, the story seemed to help the students understand what the image was. One said, "Ah... I get it" as she completed the exercise first, and correctly stated what it was.

Story did not always provide the language students needed: their *articulation* was more often poor than good for both activities. For example, this student struggled to answer the aforementioned question about words on pirate maps:

> "Depends on the amount of options / choices you have. If, there are 15 choices, there will be many "words." If there are 5 options, not as many words?"

Several students also had a difficult time expressing their ideas effectively when answering questions about the types of errors that can be made with various image representations:

> "It is much more difficult to make a mistake using image representation type (1). If a mistake occurred, the image would still be able to read the image."

> "The image is inverted, except the white showing can't acctually make up the picture. OR Everything is out of order and the image is inscifrable (sp?)"

However, good understanding was often accompanied by good articulation. Deep and good understanding tags appeared more often with good articulation than with any other tag. Similarly, poor articulation appeared with poor understanding more than any other tag.

The *representation* tags were used to capture how abstractly the students were able to think. Practical, real-world solutions were labeled "concrete", while generalization and abstraction were labeled "abstract". In both activities, students demonstrated more concrete thinking in both the observations and on the worksheets. On the Finite State Automata worksheet, for instance, few students were able to find the general patterns of the example automata, despite being able to list multiple correct examples of words included in their languages. Concrete, practical thinking was also demonstrated often for questions about the types of errors possible with the different image representations:

> "When you make a mistake with image representation type 2 is you probably miscounted and the rest of the line will be different as well."

Finally, most students were *engaged* by the stories told by the instructor. For example, the entire group was observed to be very focused on the storytelling in the Image Representation group, and 20 of the 21 study participants agreed that stories made the exercises more exciting in the post-course group discussion. In the post-course survey, two students explained why they enjoyed the Image Representation storytelling activity the most:

> "I enjoyed the story behind it. I also liked how we got to make and decode our own pixel images."

> "I liked how I didn't know what I was supposed to be drawing. It was like uncovering a secret message. I also liked the story that went with it."

In the same post-course survey, two students mentioned story when asked what they liked best about the mini-course or whether they had any other general comments:

> "It was a lot more fun then I expected and I really enjoyed the story part!!"

> "I liked doing the 'story part' as it was really easy to understand and was really fun."

Nobody mentioned disliking story in the survey, but one student did say during the discussion that she preferred to learn the topics directly rather than through the story.

Non-Story

Although there was no story, the Finite State Automata activity did include a light context of a pirate's map. Based on the results of our coding and the results in Figure 1 this group demonstrated better understanding than the story group. Though this group did not have a story to refer to, they did use the pirate map context to express their understanding as with these answers to the previously discussed question about what words in a language represent:

> "The words represent the route to find treasure island."

> "The words on the pirate's map represent different routes you can take to get to the final goal."

However, for the Image Representation activity, the non-story students displayed slightly less *understanding* than the story group. The non-story worksheet scores shown in Figure 1 tended toward less understanding than the story group's scores, though the understanding codes were distributed similarly. Without a story or context to refer to, students used technical terms to express their understanding when discussing errors in image types:

> "There will be (one) a mistaken square, one odd square out, not a row of squares (pixels) just one of them."

The non-story group demonstrated more instances of good than poor *articulation* for the Finite State Automata activity. For example, in this response about how many words a pirate's map can represent, the student gave a particularly well written answer:

> "A particular pirate's map can represent an uncountable number of words, because, in the map virtually any word will be successful in getting to the goal as long as it ends in BAB (and it was possible to get to that BAB from the previous letters before them. Theoretically, in this map a word could last for over thousands of letters if you keep going by the same islands again and again before getting to the goal."

This answer has a concrete *representation* as well, referring to the specific map the activity was based on.

The Image Representation non-story group had more instances of unusual or poor articulation. However, answers that were articulated unusually frequently exhibited good understanding. For example:

> "you may get mixed up and forget if you are supposed to leave it blank or colour it so you may end up get a drawing that needed more colouring in or shifted and not what its supposed to be."

Observations of the students in this group indicate that they were *engaged* with both activities. Two students from the non-story group chose the Image Representation activity as the one they learned the most from, and three chose it as the one they most enjoyed. Their reasons included:

> "it was interesting and something new. I enjoyed coming up with my own drawing with 1s and 0s"

> "Because the hands-on activity was entertaining and the puzzle image made me excited as to what the final answer was"

> "At first it was a bit difficult to understand but when i saw how we were supposed to do it it was really and cool."

Interestingly, another activity that used a light, non-story context but was not part of the formal study -- CS Unplugged's Usability activity involving Charlie and the Chocolate Factory -- was chosen as a favorite several times. This student explains why she liked it:

> "It was very well explained and made very much sense to me. Also, using a practical and well known example like \`Oompa Loompas' made it very enjoyable and fun!"

The Instructor Experience

Story

The instructor who told the story in class also developed its outline ahead of time. Writing the story in such a way that the lesson's content was well integrated into the plot was challenging and time consuming. Referring to Campbell's hero's journey (2008) as a starting point helped constrain the possible plots.

While telling the story, the students generally appeared interested. This was motivating for the instructor, as was hearing them say they enjoyed the story at the end (one asked "Did you think of that story?" because she liked it).

Telling the story was time consuming, however, and the instructor was unable to cover as much material as she normally does for these lessons. For example, in the Finite State Automata activity, she usually spends more time discussing the idea of the automata representing a language with words in a general way, and illustrates how a state machine might be used in games. She was only able to very briefly mention these ideas after the story was told, since it was difficult to deviate from the story to add extra detail or explore other concepts. It was also more difficult to extrapolate concepts from the way they were presented in the story and to answer questions as they arise -- the lesson is structured by the story.

Non-Story

Teaching without a story had both advantages and disadvantages. Not being bound by a story allowed the instructor of the non-story condition some additional freedom and flexibility when teaching the lessons. When students asked interesting questions that led the discussion in a slightly different direction, the instructor was free to follow those questions and relate them back to the main lesson without having to fit them into the narrative, or abandon the narrative. Not having a story also allowed the instructor to more easily transition from concrete concepts and language to more abstract and technical terms. For example, in the Finite State Automata activity, the non-story instructor was able to introduce the concept of an automaton and explain how the pirate's map was one example. She was also able to provide an example of an automaton that did not use the pirate's map, which may have helped students understand the concept more abstractly.

However, there were times where the constraint of a story would have had benefits for the instructor's thinking (as well as the students'). As part of a brainstorming session on how to encode image information more efficiently, several students made suggestions that would not work out. The instructor illustrated examples on the board of how these suggestions might not work as intended, and she found that creating examples "on the fly" without a story or context was quite difficult. Even the light context in the Finite State Automata activity helped her more easily create relevant examples.

DISCUSSION

According to our results, the students' overall experience was that they enjoyed learning through story, but did not necessarily learn better with it. While the story condition of the Image Representation activity appeared to result in better understanding than the non-story condition, the opposite was true for the Finite State Automata activity. We suggest that this is due to the abstract nature of the first activity and the pirate's map context provided in the second.

We defined *context* as a light fiction that relates a concrete scenario to a lesson and provides a vocabulary for its concepts. While the stories in our lessons contained elements like plot and character, the contexts we provided did not.

Students in the story group for the Image Representation activity benefited from the context contained in the story, but the non-story group had no similar opportunity. They had to rely on using technical terminology. This may explain why the non-story group had slightly less understanding of the topic, or at least were less able to articulate that understanding. On the other hand, both groups for the

Finite State Automata activity were able to refer to the pirate's map (automata), Pirate's Island (start state), Treasure Island (accept state), and routes through the islands (words in a language).

It is not clear what story adds over context in activities like these. We did see students making some use of story elements in their thinking, such as when the Image Representation story group recognized what the compass rose image was due to its relevance in the story. But the fact that a story has an overall structure did not appear to have additional benefit over the structure of the abstract activity done by the non-story group.

In fact, a story's structure can be a hindrance. Stories define a coherent world that can be difficult to step out of. As the story group instructor noticed, it is difficult to stop to discuss material that does not integrate well into the story, and there is less time to do so at the end of a lesson. There may be aspects of story that are more useful in subjects other than computer science. Characters and their motivations, for instance, might be employed to help a learner understand historical events. Or, we may not have yet determined how to make the best use of these elements for computer science material.

It is also important to consider that developing a story that integrates well with the curriculum takes a nontrivial amount of time and effort. Although conversations with the students suggest that they found the story-based lessons fun and exciting, our data showed they may have enjoyed activities with context just as much. There may be other ways to make use of story in the classroom that require less effort, such as using story as scaffolding on assignments and tests, or asking students to write their own stories to illustrate their understanding of a concept. These techniques might also be more appropriate for post-secondary students. Still, it is possible that using context gives you more return for the investment.

Many teachers use context and story every day in their lessons. This is particularly true in computer science, where teachers often refer to real-world problems to solve. However, it appears that we might all benefit from more careful thinking on how best to employ these devices. It may be useful to separate the possibilities into a layered model for further study, starting with lecturing about a concept, building on with an activity, incorporating context, and finally developing a story.

CONCLUSION AND FUTURE WORK

We were interested in what the student and teacher experience would be when using story to teach introductory computer science topics. We conducted an observational study with middle school girls and found that activities with story were engaging, but did not necessarily help the students learn more effectively. We also found that developing and telling a story required more time and effort than employing a simpler context to an activity. While girls in the story condition demonstrated better understanding than the non-story condition for an abstract activity, there was not a clear benefit to using story over context.

Further study could shed more light into the nature of both context and story for teaching computer science and other abstract subjects to middle school girls, and help determine under what conditions (if any) story can help students learn more easily or more deeply as compared to context. This might also aid in efforts to recruit and retain women and girls into computer science. It would also be worth exploring the use of story for other purposes; for instance, books like Lauren Ipsum are arguably more about introducing general ideas than teaching specific concepts, and story might be useful in engaging certain audiences.

WORKS CITED

Bal, M. (1997). *Narratology: Introduction to the Theory of Narrative.* University of Toronto Press, Scholarly Publishing Division.

Bell, T., Fellows, M., & Witten, I. (n.d.). Retrieved from CS Unplugged: http://csunplugged.org

Bueno, C. (2011). *Lauren Ispum.* CreateSpace Independent Publishing Platform.

Campbell, J. (2008). *The Hero with a Thousand Faces.* Pantheon Books.

Carmichael, G. (2008). Girls, computer science, and games. *SIGCSE Bull.*, 107--110.

Christensen, H. B. (2009). A story-telling approach for a software engineering course design. *Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education* (pp. 60--64). ACM.

Janit, A., Hammock, G., & Richardson, D. (2011). The power of fiction: Telling stories in abnormal psychology. *International Journal for the Scholarship of Teaching and Learning*.

Kelleher, C., Pausch, R., & Kiesler, S. (2007). Storytelling Alice Motivates Middle School Girls to Learn Computer Programming. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1455--1464). ACM.

Kubica, J. (2012). *Computational Fairy Tales.* CreateSpace Independent Publishing Platform.

Kubica, J. (2012, August). *Who is the target audience?* Retrieved from Computational Fairy Tales: http://computationaltales.blogspot.com/2012/08/who-is-target-audience.html

Papadimitriou, C. H. (2003). MythematiCS: in praise of storytelling in the teaching of computer science and math. *SIGCSE Bull.*, 7--9.

Pollard, S., & Duvall, R. C. (2006). Everything I needed to know about teaching I learned in kindergarten: bringing elementary education techniques to undergraduate computer science classes. *Proceedings of the 37th SIGCSE technical symposium on Computer science education* (pp. 224--228). ACM.

Rao, M. K. (2006). Storytelling and puzzles in a software engineering course. *Proceedings of the 37th SIGCSE technical symposium on Computer science education* (pp. 418--422). ACM.

Yarosh, S., & Guzdial, M. (2007). Narrating data structures: the role of context in CS2. *Proceedings of the third international workshop on Computing education research* (pp. 87--98). ACM.