

The Writers' Workshop for Youth Programmers

Digital Storytelling with Scratch in Middle School Classrooms

Quinn Burke
University of Pennsylvania
Graduate School of Education
Philadelphia, PA
215-990-8879
burkew@dolphin.upenn.edu

Yasmin B. Kafai
University of Pennsylvania
Graduate School of Education
Philadelphia, PA
215-746-3468
kafai@gse.upenn.edu

ABSTRACT

This study investigates the potential to introduce basic programming concepts to middle school children within the context of a classroom writing-workshop. In this paper we describe how students drafted, revised, and published their own digital stories using the introductory programming language Scratch and in the process learned fundamental CS concepts as well as the wider connection between programming and writing as interrelated processes of composition.

Categories and Subject Descriptors

K.3.2 [Computing & Information Science Education]:
Computer Science and Education, Curriculum, Literacy

General Terms: Design, Human Factors, Languages

Keywords: Computer Science education, Scratch,
programming, digital storytelling

1. INTRODUCTION & RATIONALE

With the current efforts to broaden participation in computing and introduce computational literacy on the K-12 level [19], there is a need for educational approaches and models that connect to existing curricular practices. According to the recent report from the Association of Computing Machinery, *Running On Empty* [24], less than two-thirds of K-12 schools in the country offer any form of standardized CS-based curricula. While organizations like the ACM and Computer Science Teachers Association (CSTA) continue to push for such standards on the state level, this study takes an alternative route to getting CS more immediately into the classroom. Building upon previous research teaching programming in terms of storytelling [3; 15], this paper introduces the writers' workshop model [4] as a means to facilitate a particular *process* by which youth can learn programming during the school-day and within core-curricula subject matter.

The widespread growth of the writing workshop on the K-12 level can be traced to the seminal publication of Calkins' [4] *The Art of Teaching Writing*. Promoting a communal setting over solitary endeavor and stressing writing as a perpetual process and not

Simply the finished product, the writing workshop opened composition as a form of personal expression available to all children. Such a shift was nothing less than a sea-change on the K-

12 level, as over the vast majority of the 20th century, writing was a discipline in which the finished product garnered far more attention than the process by which it was created [21]. Yet as Calkins' own mentor Donald Murray cautioned educators about the craft: "Writing might be magical—but it's not magic. It's a process, a rational series of decisions and steps that every writer makes and takes, no matter what the length, the deadline, even the genre..." [10]. Writing does not simply magically appear nor is it an elite skill limited to a select few based on gene-pools. It is a learned process that can always be further honed and developed through personal reflection, endeavor, and shared experience.

Much like writing three decades ago, computer programming still faces this myth of the "magical". Studies [14; 18] suggest, one of the primary reasons for the declining enrollment and lack of diversity within CS as a major is based upon this perception that the field is meant for only a select few who happen to be inherently skilled at it. As DiSalvo and Bruckman [8] point out about their recent efforts to broaden and diversify student interest in the field, "computer science itself is not that difficult—but wanting to learn it is" (p. 27). Efforts to demystify programming through activities like game design [7] and storytelling [15] have been successful in introducing children to CS at earlier ages. However, because these approaches' main emphasis have been centered upon the learning of programming (with game design and storytelling playing only a secondary, "prop" role), they have been relegated to afterschool or summer club activities in which children self-select to participate (often based upon a pre-existing interest in learning computer science). Meanwhile within the classroom, the overwhelming majority of K-12 student continue to remain only vaguely aware of programming as a utilitarian tool, while educators remain wary of attempting to introduce yet another subject into curricula already stretched tight by high-stakes test prep and accountability measures.

Here we propose a different approach that leverages some of the previous successes we have had using storytelling to teach programming [3] but now brings to bear particular alignment to state standards in language arts instruction. Our study takes place in the context of a school-day classroom using the structure of a writing workshop as a means to (a) facilitate a deliberate process by which children can learn computer programming, and (b) leverage the professional knowledge of K-12 educators in traditional English/ language arts classrooms to better integrate programming into core-content classroom activities. With a class middle school students, we conducted a two-month long writing workshop that focused on generating one's own digital stories using the programming language Scratch. Our focus was two-fold: first, to what extent can the existing English/ language arts frameworks be used to tie the composition process to digital storytelling in Scratch? And second, how do the Scratch designs incorporate both narrative and programming elements and to what extent do students can students appreciate such overlap?

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGCSE'12, February 29–March 3, 2012, Raleigh, North Carolina, USA.
Copyright 2012 ACM 978-1-4503-1098-7/12/02...\$10.00.

2. BACKGROUND

The pedagogical underpinnings of the writers workshop draw on extensive research from community of practice [16] and communities of learners [2] in which individuals develop their composition skills based on three crucial elements: (1) authentic practice; (2) finished product, and (3) collaborative support. In his review of literature focusing on the connection between learning programming and general problem solving skills, Palumbo [20] likewise stresses the need for K-12 programming instruction to be developed through a series of stages, in which the task is meaningful to participants, and feedback is mutual and continuous. While early introductory programming languages such as Logo had success in entering school classrooms in the early 1980s, such success was short-lived in part due to a lack of authentic practices and products which tied such learning to wider problem solving skills.

2.1 Authentic practice

In “Thick Authenticity: New Media and Authentic Learning”, Shaffer and Resnick [23] posit that digital media has the opportunity to create learning environments for children that are at once (1) personal, (2) real world, (3) disciplinary, and (4) assessable. However, too often instruction using computers and digital media is simply “teaching computers” as opposed to teaching through the computer. Consequently while children may have in-depth knowledge of a wide range of digital media applications, there may be a severe lack in understanding how such media relate to each other and to oneself for personal expression. Likewise, in terms of writing, learning parts of speech and the structure of sentences without any designated purpose beyond grammar acquisition fails to produce effective writers [17]. The writing workshop provides an alternative environment to learning composition, in which narratives are generated not based upon knowledge acquisition but upon personal reflection and individual expression.

2.2 Finished product

Learning through-design ties back to project-based learning which itself is based upon the Constructionist model in which students simultaneously learn new information and design a product which reflects such learning [11; 12]. Previous studies exploring storytelling and programming through the constructionist model [15; 26] focus on using stories as a means to make coding more accessible and palpable to children. However in both cases, the value of storytelling as the finished product was only considered in terms of its ability to interest children in programming, and there was little consideration how the story genre offered a viable link between the discipline of programming and the discipline of writing. While code is certainly a valuable skill, using digital storytelling simply as a way to draw kids into programming neglects to take into account the full and rich ways such storytelling can also be used to develop children’s sense of narrative structure. As Sandy Hayes points out, “Students don’t have to produce standardized writing to meet writing standards” [10]. Programming-as-storytelling in the setting of a writing workshop represents one such potential “unstandardized” format that deserves further exploration in schools.

2.3 Collaborative support

Black’s research [1] on teens’ fan fiction writing suggests that when youth can find an environment where writing acts as a social

outlet, a collaborative process, and a means for personal expression, their output can be both prodigious and notable in both style and content. Writing—a process which K-12 schools still regularly struggle to make creative, personal, and collaborative—took on a far more interactive nature in a forum outside of the classroom. Our study here builds directly on Black’s research, examining the various multi-media projects children share and comment upon on another file-sharing website, <http://www.scratch.mit.edu>. While not traditional pen-and-paper compositions, the programming projects children write and share using the Scratch website are very much digital “texts” incorporating words, images, and sounds to produce a wide variety of stories, games, and animations. Students not only have the opportunity to share their digital stories with each in the workshop but within the wider Scratch community, which currently has over 800,000 registered members and nearly 2 million uploaded projects.

In terms of combining authentic practice and collaborative support with a constructionist learning model, we find that a number of other recent efforts such as Glitch testers [8] and Scratch-based Collaboration Challenges [13] use these same principles in setting up successful educational projects. Yet what we are proposing with the writing workshop for programmers is a K-12 pedagogical model that has potential to introduce coding in core academic subject matters, align with state standards.

3. The Writers-Workshop for Middle School

Participants

For seven weeks in the Fall of 2010, we set up 11 writing workshop sessions in an elective course using Scratch at an urban public middle school located in West Philadelphia. Ten students—all boys, ages twelve to fourteen—participated in the writing workshops and were representative of the schools’ diverse population of African-American, Caucasian, and Latino children. In total, we collected eleven projects (one participant created two stories) by the end of the program.

Choice Elective & Alignment with State Academic Standards

Over the course of the seven weeks, every Choice session would open with a brief “mini-lesson” [5] emphasizing a particular element of effective composition (such as characterization, foreshadowing, setting a scene) which would likewise be tied to learning a particular coding procedure in Scratch (e.g., using the broadcast feature to establish dialogue, importing external images, using loops to standardize behavior). Every mini-lesson was supported by anywhere from one to three sample digital stories selected from the Scratch website, which exemplified a particular storytelling element or genre of storytelling (e.g., mystery, action/adventure) featured within the lesson. This not only grounded the lessons in practical application but offered an excellent segue to examining the actual coding scripts of the projects, exploring exactly how the sample story creators achieved a particular effect with the Scratch programming language. All lesson plans were aligned to Pennsylvania state standards Reading, Writing, Listening, and Speaking on the 8th grade level and supported by the school’s junior-high literacy instructor Mrs. Steinberg, who offered feedback rubrics and pre-writing activities from her own classes which were based on her use of Calkins’ [5]text.

	Subject:	Grade(s)	Course
TU. 11/9	Content Area: ELA	5 th - 7 th	Scratch Choice
	Standards: 1.9.7.A, 1.9.7.E		
	Obj: The objective for the initial session is for kids to (a) get acquainted with the basic features of Scratch & (b) start generating ideas for their own stories		
	Materials: Laptops w/ Scratch		
	Pre-Class Activity: Kids should enter the club & begin getting out to laptops, accessing the software		
	Lesson: After introducing yourselves & your backgrounds, focus on initially		
	<ul style="list-style-type: none"> > Introducing the 3 panels of Scratch & getting a sample sprite to move & speak > Ensuring the kids have Scratch as a download on their laptops > Briefly discuss creating stories w/ Scratch & demo 2 sample projects 		
	<ul style="list-style-type: none"> o My Scratch Story (traditional) <ul style="list-style-type: none"> http://scratch.mit.edu/projects/idigcows/123644 o Harry Potter & the Monster Under the Bed (much more intricate) <ul style="list-style-type: none"> http://scratch.mit.edu/projects/mizzzy42/11516 		
	Activities: start playing with Scratch		
	Writing Activity:		
Closure: Review the basic 3 panels over the final 5 mins. & introduce the Scratch website (http://scratch.mit.edu) for those who are able/ would like to work from home			
Assessment: No formal assessment but take note—are kids paying attention? Engaged?			
Assignment: Start thinking about your own story ideas—come in with 2 ideas for next week!			

Figure 1: Lesson from the outset of the workshop, created within the template recommended by the Philadelphia School District and aligned to PA academic standards

The workshop followed five stages over the seven weeks, which, while distinct, did have some overlap from week to week depending on individual student progress:

- **Pre-writing/ Planning (Weeks 1-2):** Every participant generated 3-4 “seed ideas” [5] and entered these into their Writer’s Notebook, which they then reviewed with us for feedback.
- **Drafting (Weeks 2-3):** Once students had discussed their seed ideas with us, they proceeded to sketch out their ideas using storyboards. Using a pencil, kids drew out their individual shots with the knowledge that these screen-by-screen renderings would act as a “roadmap” for their compositions.
- **Revising (Weeks 3-6):** Once their Storyboards had officially been approved (sessions 3 & 4), the middle-schoolers began to compose their actual digital stories. All participants utilized both a “bottom up” and “top-down” approach to composing their stories, in which the former refers to creating anew in Scratch, while the latter involved sampling others’ projects and repurposing the code for their own projects. The majority of participants leaned more to “bottom up” composition, particularly over weeks 3-4 of the *Choice* class.
- **Editing (Week 6):** The briefest stage of the 5, students made final revisions based on comments they had received online as well as during weeks 6-7 of class. Many of the edits were simply “fine-tuning” in terms of correcting spelling and grammar in characters’ dialogue or trouble-shooting the programmed behavior of a coded sprite.
- **Publishing (Weeks 6-7):** All students posted to the Scratch website again over the final two weeks of the workshop; over the final day, students presented their final projects to their classmates in terms of plot and characterization as well as in terms of the underlying code operating their digital stories.

4. DATA ANALYSIS

During the workshop, we collected a variety of data sources:

CS attitudes pre/post surveys: Adapted from a Georgia Tech’s computer science attitudes survey developed by Lijun Ni and Mark Guzdial (<http://coweb.cc.gatech.edu/mediaComp-teach/16>), pre-and post-surveys were given to all participants in the study gauging their familiarity and attitudes to digital media, their sense

of their own storytelling and computer capabilities, and their own attitudes toward working collaboratively and creatively.

Field Note Observations: Collected daily and transcribed within a twenty-four period, they were subsequently coded thematically capturing particular usage trends across the workshop.

Video Footage: All sessions were videotaped in their entirety with select sections transcribed for the sake of better capturing a moment-by-moment understanding of how students use the software in the workshop. All post-session interviews with students were video-taped and transcribed.

Scratch Project (artifact) analysis: All Scratch projects were periodically collected over the duration of the program (a minimum of three times per project) and subsequently examined in terms of their staged storylines and underlying coding scripts. All projects were also analyzed in regard to their programming blocks using *Scrape* technology, a tool developed by RiverSound Media (<http://happyanalyzing.com>).

Post-Interviews: At the program’s end, all students participated in 5-10 minute interviews gauging their experience in the classroom. As with the field note observations, these interviews were subsequently coded thematically.

5. FINDINGS

Over the course of the workshop, students learned both the fundamentals of programming and storytelling, and this is charted here in terms of the products (digital stories) they programmed, the processes (debugging and revising) they utilized, and their overall perceptions of the workshop at its close.

5.1 Product

In terms of product, 9 out of the 10 participants generated a complete digital story, entailing multiple characters, settings, and plot stages. Each finished project also entailed a number of key coding concepts not simply characteristic of Scratch but of all programming languages—from Java to C++—clearly indicating that over the workshop’s eleven sessions the middle school students not only composed their own digital stories in the software but also learned and applied some key fundamentals programming concepts in the process.

The chart that follows highlights some of the programming concepts students used in the creation of their stories as well as the frequency of use:

Table 1: Types & frequency of programming concepts utilized

Programming Concept	% of Projects Utilizing the Concept	Frequency per Project
Coordination & Synchronization	100%	8.8
Threads (Parallel Execution)	100%	1.6
Loops	90%	2.5
Event-Handling	100%	1.5
Boolean Logic	20%	1.6
Conditional Statements	30%	1.7
Variables	10%	.3

One such story was Darryl’s crayfish tale (below) which was based on a real-life experience in which he won a salt-water crayfish at school but ended up killing the poor animal with tap water.



Figure 2: Darryl’s whimsical project “Crayfish” depicting the creature’s untimely end

Though Daryl’s digital story is brief (approximately a minute-and-half long), it took over ten hours to program and the coding scripts he utilized were very much characteristic of those of his peers in the classroom. Like the wider class, Daryl relied heavily on the use of costume changes and dialogue between characters to propel his story forward. Also like his peers, Daryl’s project made use of a single input key (in his case, the green “go” flag in the upper right-hand corner), which once clicked, set the story in motion to its conclusion. Unlike a number of his classmates, Daryl’s project demonstrated more intricate programming through his use of the “broadcast” command which allowed for his story’s programmed objects to trigger the behavior of other objects, precluding the need of timed intervals to coordinate events.

5.2 Process

These commonalities among projects in terms of code are not unsurprising given that all students followed the same process in the classroom, learning code through storytelling. All ten participants had a strong sense of the stages of writing. This was the expectation—having met with Mrs. Steinberg the month prior to the workshop, she assured me the students went through these stages of composition whether they were composing a poem, a graphic novel, or an expository essay. However, even more than their familiarity with the stages of writing, the middle-school students’ familiarity with the elements of writing—particularly the elements of creative fiction—offered a particularly useful scaffold as means to introduce the basic elements of Scratch over the first two weeks of the workshop. Couching the use of Scratch in terms of common language arts concepts such as rising action and static versus dynamic characterization proved remarkably effective in acquainting (or in some cases, reacquainting) the students to the Scratch programming language. Simply being able to describe basic elements of programming in writing terminology eased participants’ unfamiliarity with the coding process.

Table 3: Addressing the CS language “barrier” through analogous terminology

Writing Term	Programming Term
Protagonist/ Antagonist	“sprites”
Drafting	“design”
Revising	“debugging”
Character Motivation	coding “parameters”

“Oh yeah, I understand ‘round’ versus ‘flat’ characters,” Marcus remarked with some surprise when I explained during session #3 how flat “stock” characters’ programmed behavior could be “looped” while the protagonist’s more dynamic (and thus “round”) behavior would be far less repetitive. Accordingly, children learned to program Scratch sprites based on a particular character’s motivations. In this sense, literary elements such as characterization and setting served not only as a means to introduce programming terminology, but also acted as the vehicle through which children learned how to program. In the case of 8th grader Marcus’ fantasy basketball game, protagonist and NBA star Tim Hardaway had a diverse, linear-based coding sequence, timed out in intervals. Meanwhile antagonist LeBron James’s coded sprite (below) had limited programmed behavior that was far less varied and entirely looped, typical of characterization that is both flat and static in nature.

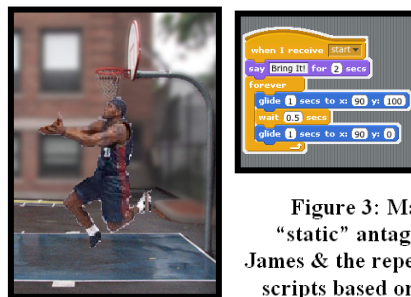


Figure 3: Marcus’s “static” antagonist LeBron James & the repetitive coding scripts based on his motivation

5.2.1 Brainstorming & Outlining

As indicated in the planning guide and sample lesson above, students began to map out their digital stories in Scratch over sessions #3 and 4 of the workshop. Generally, students relied on three different sources to generate ideas for their potential digital stories (none of which were mutually exclusive):

- **writer’s notebook:** distributed by Mrs. Steinberg to every 7th and 8th grader, the black-and-white speckled pad is the mandated starting point for any composition in her classes. Intended as a place to simply write down ideas, students need to generate at least three potential ideas before they opt for any single one—a requirement which was maintained for the workshop as well.
- **Sprite cache:** an assortment of various character images, ranging from people to animals to alphabet letters are stored within the Scratch software; users can click upon these stocked folders to import various sprites into a project.
- **popular culture:** while the term “popular culture” encompasses an innumerable array of source-material, here it refers to those images that students searched out over the Internet, saved to their laptops, and imported into Scratch as image files (JPEGs typically); learning this process over session #3, many students grew very excited as it allowed them to utilize figures from their own favorite stories and games as their lead characters.

While students eagerly imported and tinkered with various Sprites from both the web and Scratch caches over the first two weeks, getting them to commit to a set narrative via the storyboards proved to be a real challenge, though this challenge was not necessarily an unexpected one. Previous studies utilizing storyboards for Scratch-based narratives [3; 6] likewise found some measure of children’s resistance to using storyboards, partially due to an unwillingness to commit to a single narrative

and partially due to participants' reluctance to return to pencil-and-paper after having begun to compose digitally. "Aw, come on, I already know what I want" protested Darrell over session #4 upon learning that all participants needed a completed storyboard before proceeding with their digital stories. While Darrell already felt confident he knew his narrative pathway without help from the storyboard, other participants were uncertain about which captions they should sketch out as the key passages to their storylines. And still others—despite repeated mollification that the storyboards were a mere outline—worried their pencil-and paper renderings were going to be critiqued for a lack of artistry. Yet, despite these issues, as expected, the storyboards did get all of the students actually verbalizing their ideas aloud through the process of putting them to paper, and they did provide a discernible end-point to the composition process—namely, the resolution of the narrative. All participants had completed and submitted their storyboards by session #5. The shortest storyboard consisted of a mere 3 captions while the longest extended to 7 captions, with 5 captions as the most common length.

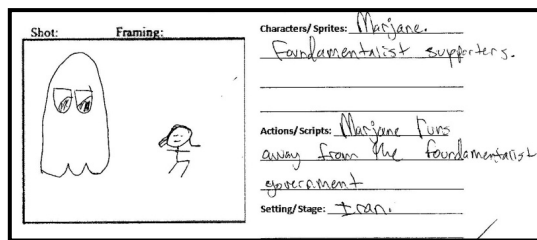


Figure 4: Sample caption from Carlos' Scratch version of *Persepolis*

5.2.2 Drafting, Feedback, & Revising

As an outline, the storyboards served as the student's raw "roadmap" and was the first piece they submitted and received formal feedback by way of written comments on the paper itself. While three of the storyboard submissions were fairly perfunctory, including only an absolute minimal amount of detail, the remaining seven were well-organized and well utilized the side-space alongside the caption box to explain *the who?* (character/sprites), *the what?* (actions/ scripts), and *the where?* (settings/stages) of each progressing scene.

Following these notes on the storyboard submissions, feedback over the next three sessions was more informal, including comments and suggestions on individual student projects as the group worked independently on them. The entire class participated in ten-minute "gallery walk" midway through the workshop, leaving their laptops open to their in-the-works projects and then walking the room with their peers, sampling each others' stories and asking questions based on what they viewed thus far. To a degree, the projects students had prepared for the gallery walk served as their initial drafts; however, no participant had actually completed his digital story at this point, which made giving constructive feedback more difficult for the students. Students largely commented on the appearance of each other's characters (e.g., "cool costume", "nice look") but had a difficult time providing more substantial feedback about elements like plot development and characterization. "So what's supposed to happen here?" 8th grader Todd asked of his friend's Greg's project during the gallery walk, unable to offer much more given that he was entirely uncertain where the narrative was actually heading.

More directed feedback came from us as instructors during the next session when all students posted their draft projects at the

Scratch website. Using the anonymous username "SLA_User" and with storyboards in hand, we reviewed each project based upon what had been uploaded to the Scratch website thus far and what the remaining captions on the storyboard indicated should happen next. Using the "Comments" feature on the website, we posted brief observations, small items of encouragements, and occasional questions. Given the character-limit of the Comments box as well as the decidedly "non-academic" nature of the website, we opted to keep the comments succinct and casual; the goal was not to exhaust the students with a "to do" list but rather engage them with the prospect of sharing their work with wider audiences online.

5.3 Perceptions

Based on the post-survey, 70% of respondents agreed or strongly agreed that the storyboard helped them create their stories; 70% agreed or strongly agreed that they learned more about computing during the workshop, while 80% indicated they learned more about storytelling during the workshop; 70% agreed or strongly agreed that they felt better at computing based on the workshop; 90% agreed or strongly agreed that they had enjoyed the workshop experience; and 70% agreed or strongly agreed that anyone can be a good storyteller in Scratch if he or she works hard at it.

In post-interviews, it was storytelling that occupied much of the one-on-one feedback from students, with multiple middle-schoolers stressing the importance of the storyboards in ensuring they had a particular idea in mind for their Scratch project. "Scratch can do almost anything," explained Daryl in his post-interview, "It has hundreds of controls, hundreds of images and you can even take ones of the Internet... And so, all you need to do is have a focus."

6. DISCUSSION

Returning to our research questions, it is clear that the writing workshop setting alongside the school's existing language arts standards proved to be not only an effective framework for facilitating middle school children's digital composition within Scratch, but also underscored the wider connection between coding and writing as interrelated processes of composition. Digital storytelling in Scratch—particularly in terms of the workshop's focus on characterization and plot analysis—offers a new medium through which children can exercise the composition skills they learned within traditional literacy classrooms while also offering the mutual benefit of introducing coding at earlier ages.

However, as evident in Table #1 above, certain coding bricks in the storytelling workshop—namely, conditionals, Boolean Logic, and Variables—were not widely used. These coding scripts are characteristic of games in which there are no fixed outcomes, and such scripts are simply not integral to linear narratives. Therefore participants did not learn much about these essential coding concepts over the seven weeks of the workshop. This finding is not an indictment of the usage of storytelling to introduce programming, but rather a caveat that while the writing-workshop model offers certain advantages in introducing coding to children, its products only represent the end-result of a certain type of programming. Moving forward, we are interested in expanding the parameters of the workshop to have youth compose interactive stories, which will not only introduce a wider range of programming variables but offer further insight on the intersection of narrative and game-making in digital media [22].

In terms of future workshops, there is also very much the need to explore (and make more explicit) the intersection between coding and narrative composition as they relate to computational thinking in the classroom [19; 25]. The workshop we designed and enacted largely held storytelling and coding apart as separate entities—each integral to and serving the same end goal—but introduced separately nonetheless. Computational thinking, which emphasizes the practical and *creative* functionality of algorithms, offers a potential new lens for accentuating the connection between coding and writing, both of which attempt to articulate a precise input in order to facilitate a particular output. *Make* magazine’s recent partnership with the National Writing Project [9] is based upon such a premise, creating a series of workshops for literacy teachers that emphasize their role as “makers” within the classroom and student writing as a tangible “product” in which the technical and creative are inextricably intertwined. As systematically coded images and sounds placed in aesthetic juxtaposition, digital stories in Scratch are likewise “products” that embody both the technical and the creative elements of composition and offer a broader conception of what “writing” with computers may look like in the 21st century.

This said, while it is crucial to broaden K-12 schools’ conception of literacy as well as make computer science more interdisciplinary in nature, it is equally important to be mindful of the respective disciplines out of which this workshop originated and the potential limitations of our study for instructors in these disciplines. In the discipline of English/ language arts, literacy instructors first and foremost need the designated time to enact such a weekly workshop, as well as a population of students familiar enough with the stages of writing to be capable of comfortably following them in terms of Scratch storytelling. In the discipline of computer science, CS instructors may well appreciate the Constructionist nature of the workshop, yet there still needs to be a continued push to ensure that programming enters schools primarily through statewide academic standards specific to CS (and not simply technology). For them, this study represents an alternative—and hopefully only temporary—route to getting CS into the K-12 classroom.

7. REFERENCES

- [1] Black, R.W. 2008. *Adolescents and Online Fan Fiction*. New York: Peter Lang.
- [2] Brown, A., and Campione, J. 1994. Guided discovery in a community of learners. In K. McGilly (Ed.), *Classroom Lessons: Integrating Cognitive Theory and Classroom Practice* (pp. 229-290). Cambridge, MA: MIT Press.
- [3] Burke, Q. and Kafai, Y.B. 2010. Programming and storytelling: Opportunities for learning about coding and composition. In *Proceedings of the 9th International Conference on Interaction Design and Children Conference 2010* (Barcelona, Spain, June 09-11), pp. 348-51.
- [4] Calkins, L. 1986. *The Art of Teaching Writing*. Portsmouth, NH: Heinemann.
- [5] Calkins, L. 1994. *The Art of Teaching Writing* (2nd Ed.). Portsmouth, NH: Heinemann.
- [6] Costa, N. 2009. Animating Aesop's Fables. *Teachers Network*. Retrieved August 12, 2011 from <http://teachersnetwork.org/grantwinners/Teachnet09NancyCosta.htm>.
- [7] Denner, J., Werner, L., Bean, S. and Campe, S. 2005. The girls creating games program: Strategies for engaging middle-school girls in information technology. *Frontiers: A Journal of Women Studies*, 26, 1, 90-98.
- [8] DiSalvo, B. and Bruckman, A. 2011. From interests to values. *Communications of the ACM*, 54, 8 (Aug 2011), 27-29.
- [9] Dougherty, D. 2010. Teachers as makers. *O'Reilly Radar* (December 11, 2010). Retrieved August 23, 2011 from <http://radar.oreilly.com/2010/12/teachers-as-makers.html>.
- [10] Feinberg, B. 2007. The Lucy Calkins Project. *Education Next*, 7, 3, 26-31.
- [11] Harel, I., and Papert, S. 1990. Software design as a learning environment. *Interactive Learning Environments*, 1, 1-32.
- [12] Kafai, Y. B. 2005. The classroom as living laboratory: Design-based research for understanding, comparing, and evaluating learning science through design. *Educational Technology*, 45, 28-33.
- [13] Kafai, Y.B., Fields, D.A., Burke, Q. 2011 (accepted). Collaborative agency in youth online creative production in Scratch. In *Proceedings of the 19th International Conference on Computers in Education* (Chiang Mai, Thailand, Nov 28 – Dec 2).
- [14] Kelleher, C. and Pausch, R. 2005. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, 37, 2, 83-137.
- [15] Kelleher, C. and Pausch, R. 2007. Using storytelling to motivate programming. *Communications of the ACM*, 50, 7, 59-64.
- [16] Lave, J., and Wenger, E. 1991. *Situated Learning: Legitimate peripheral participation*. Cambridge, England: Cambridge University Press.
- [17] Lensmire, T. 1994. *When Children Write: Critical Re-visions of the Writing Workshop*. New York: Teachers College Press.
- [18] McGrath-Cohoon, J. and Asprey, W. (Eds.) 2006. *Women and Information Technology: Research on Under-representation*. Cambridge, MA: MIT Press.
- [19] National Research Council. 2010. *Report of a Workshop on the Scope and Nature of Computational Thinking*. Washington, DC: National Academies Press.
- [20] Palumbo, D.B. 1990. Programming language/problem-solving research: A review of relevant issues,” *Review of Educational Research*, 60, 1, 65-89.
- [21] Resnick, L.B. 1987. *Education and learning to think*. Washington, DC: National Academy Press.
- [22] Robertson, J. and Good, J. 2005. Story creation in virtual game worlds. *Communications of the ACM*, 48, 1, 61-65.
- [23] Shaffer, D. W., and Resnick, M. 1999. Thick authenticity: New media and authentic learning. *Journal of Interactive Learning Research*, 10, 2, 195-215.
- [24] Wilson, C., Sudol, L., Stephenson, C., and Stehlik, M. 2010. *Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age*. ACM. DOI = <http://www.acm.org/runningonempty/fullreport.pdf>.
- [25] Wing, J.M. 2006. Computational thinking. *Communications of the ACM*, 49, 3, 33-35. Texas
- [26] Wolz, U., Ouyang, Y., Leutenegger, S.T. 2011. Scratching the subject surface. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*. (Dallas, TX, March 09-12). ACM, New York, NY, 48-90.