INTERNATIONAL LITERACY ASSOCIATION

Debugging the Writing Process: Lessons From a Comparison of Students' Coding and Writing Practices

Ziva R. Hassenfeld, Marina Umaschi Bers

This study of three young students' compositional activities explores how coding can support the teaching of writing.

A Story Emerges

Sarah looked intently at the picture of a cartoon turtle and fox hugging. She tapped her pencil against her writing journal and stuck her tongue out slightly. Soon, with a spark of inspiration, she began to write. She wrote in silence, her head buried in her writing journal. Five minutes later she looked up. "Done," she said (see Figure 1): "Tirtil woke up and wantito take a wouk. So he did in the forest. In the forest he met a fox and became frend." Ziva (first author) asked Sarah to read her composition out loud, curious whether rereading her composition would encourage her to review and revise. Sarah reread her story out loud, but she did not change anything.

A week later, Sarah returned to the same corner of the classroom and continued working on her Fox and Turtle story. After six minutes she declared herself done (see Figure 2):

> Tirtil woke up and wantitto take a wouk. So he did in the forest. In the forest he met a fox and became frend. but they did not now woat to play so they decided to take a waok and they meet lion and lion wanted to play tag and they all playd happly ever after the end.

She reread her composition and, again, declined an invitation to revise.

The following week, Sarah was in the same corner with the same Fox and Turtle story. This time, Sarah read what she had composed in the previous two weeks and declared, "That's a weird story." She pointed to a few words. "This is not how you spell this stuff," she reflected. Then, instead of addressing the problems in her writing that she had noticed, she meticulously crossed the whole thing out—line by line, page by page. She began again (see Figure 3):

> one day tiger decided / he wonted to take a / walk. on his walk he met up with / turtle — go on go on / (turtles mom says) turtle time for diner / tigers mom says tiger / time for diner the end.

In Sarah's new composition, gone was Lion. Gone was the story of a burgeoning friendship. Gone, too, was the fairytale ending, replaced with their mothers calling the animals in for dinner. Fox had become Tiger, and apparently, now the characters were young. The next time Sarah and Ziva sat down together to write, Sarah moved on to a new story, leaving a myriad of questions about "Fox and Turtle" unanswered.

This portrait of a 7-year-old's writing process feels familiar to anyone who has worked with young writers. Their writing can be incomplete and their appetite for revision limited (Applebee, Langer, & Mullis, 1986). This is, after all, understandable. Writing is hard.

Writing Is Hard

Students come to writing with years of experience in speaking. However, learning to write poses a new set of challenges (Nystrand & Himley, 1984). Familiar with the interactional structure of speech, students

Ziva R. Hassenfeld is a postdoctoral fellow in the Eliot-Pearson Department of Child Study and Human Development at Tufts University, Medford, MA, USA; email ziva.hassenfeld@tufts.edu.

Marina Umaschi Bers is a professor in and the chair of the Eliot-Pearson Department of Child Study and Human Development at Tufts University, Medford, MA, USA; email marina.bers@tufts.edu.

 doi:10.1002/trtr.1885 © 2019 International Literacy Association

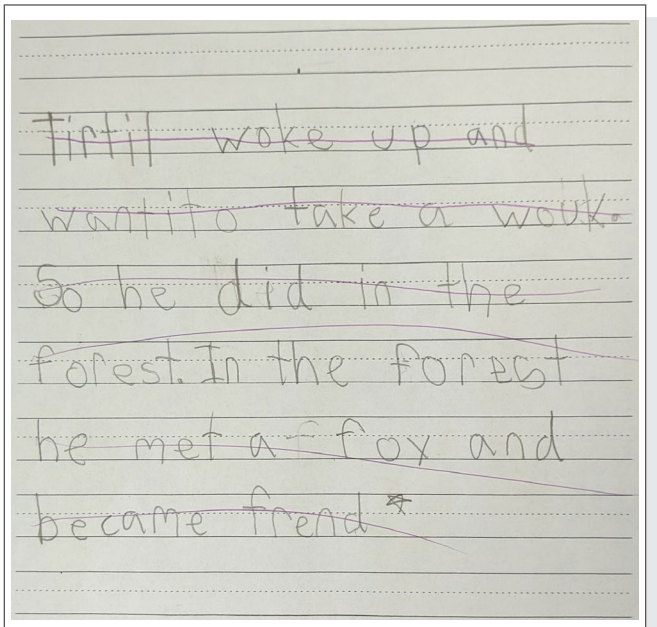
must learn a new structure for writing that relies on a clear order and sufficient detail. Used to speaking in clauses, students must learn to write sentences, a new syntax that belongs exclusively to writing. Used to the concrete motivations for speech, direct communication, and needs, students must find inspiration in a more abstract motivation to write at all (Dyson, 1989; Kress, 1994).

From the student's immediate perspective, writing requires a lot of energy with little obvious return. Writing seems to be nothing more than speech with extra burdens. As one student put it to us, "I don't like writing if it's too much to write. I mean, I like thinking of stories, but it's annoying when I have to write it." For this student, like the other students we worked with, the value of writing was not clear. If writing is just speech plus mechanics and conventions, then what is the point? Yet, learning to write is a primary instructional imperative for schools. For decades, researchers and practitioners have puzzled over how to make writing instruction more effective.

PAUSE AND PONDER

- What does the process of coding share with the process of writing?
- How can learning coding support students in practicing elements of the writing process?
- Are there authentic ways to integrate coding and writing in the classroom?

Figure 1
Sarah's First Written Composition

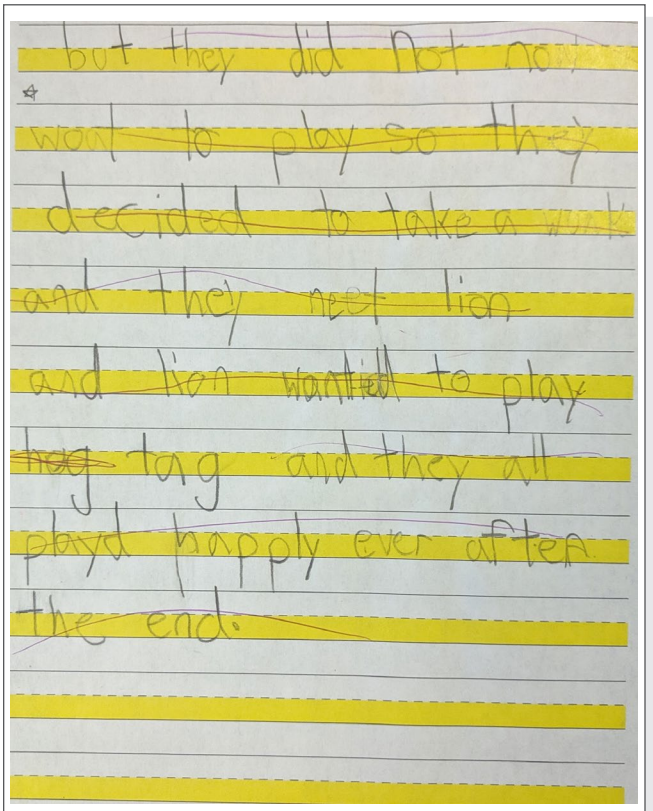


Note. The color figure can be viewed in the online version of this article at http://ila.onlinelibrary.wiley.com.

Reading, Writing, and Coding

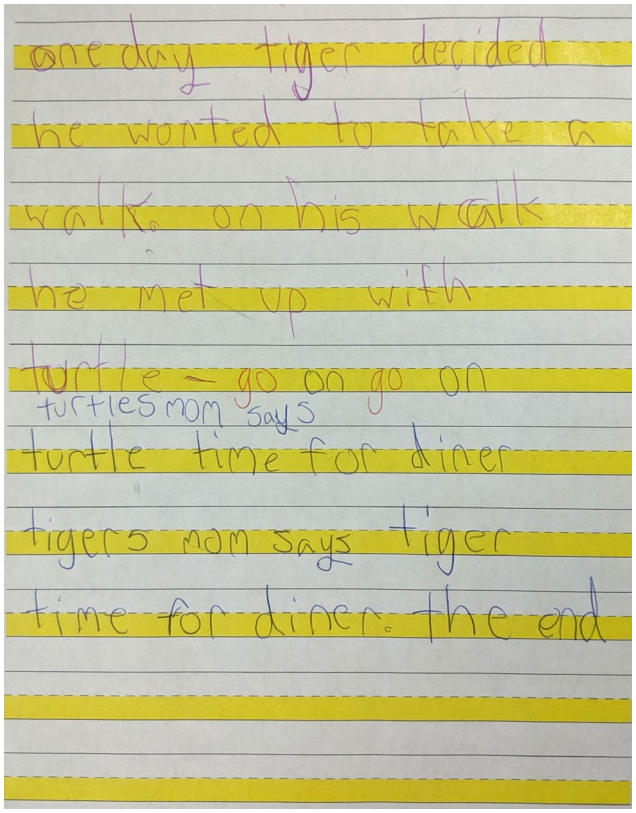
One pathway toward understanding effective writing instruction has been to compare it with other activities that students study in school. In their article "The Author's Chair," Graves and Hansen (1983) argued that reading and writing share a fundamental structure; they are both "composing processes" (p. 180). Through reading, students become aware of the choices that authors make. They begin to imagine themselves as authors and writing as a relatable, even enticing activity. As students reflect on the decisions that authors made in books, students begin asking questions of themselves as authors: Would I make that choice? How would I begin my story? and so on.

Figure 2
Sarah's First Written Composition, Continued



Note. The color figure can be viewed in the online version of this article at http://ila.onlinelibrary.wiley.com.

Figure 3
Sarah's Second Written Composition



Note. The color figure can be viewed in the online version of this article at http://ila.onlinelibrary.wiley.com.

Hansen (1987) continued to argue that because of their fundamental relation, reading and writing should be taught as two aspects of an integrated whole. Attention to the connection between reading and writing has remained a focus of literacy research in the decades since (see, e.g., Eckhoff, 1984; Langer & Flihan, 2000). In this article, we propose that it now makes sense to extend this notion of composing processes to coding as well.

In recent years, scholars have studied the comparative study of writing instruction to take into account digital developments. Scholarship on multimodal texts has examined the various new virtual canvases on which, and in which, traditional reading and writing take place (see, e.g., Kress, 2010; Leu et al., 2008; Marsh, 2011). Scholarship on new literacy has theorized the ways in which coding is changing the activity of reading and writing (see, e.g., Gee, 2003; Kress, 2003; Leu, Kinzer, Coiro, & Cammack, 2004; Vee, 2017). Research has yet to look at coding side by side with reading or writing among emergent composers.

Ziva and Marina (second author), both researchers—one from the world of literacy and the other from the world of computer science education—came together to explore the compositional processes (Graves & Hansen, 1983) of beginning writers and beginning computer programmers. In this study, we explored young students' writing processes alongside a second compositional activity that many students are now learning in elementary school: coding. After a brief framing of the theoretical construct that situates writing and coding in the same pedagogical conversation and an overview of the context in which this research took place, we will explore how the students composed in these two modalities and provide suggestions for how coding and writing can be integrated in the curriculum.

Compositional Processes Expanded

Writing and coding are both compositional processes that share a subset of activities (see Table 1). Exploring these activities in each domain, writing and coding, and how they manifest in young students' work begins to fill in the conceptual picture of reading, writing, and coding as compositional processes that can support one another.

Planning and Prewriting

In coding, *planning* refers to the process the programmer goes through before she begins programming the actual composition or project. It can involve creating a flowchart or completing a design journal (Strawhacker & Bers, 2015). In writing, *prewriting* refers to everything that happens before the writer begins the actual composition. It can involve doing research on a topic, outlining one's composition,

Table 1
The Coding Process and the Writing Process

Coding composition	Writing composition
Planning	Prewriting
Creating	Drafting
Testing: Evaluating	Evaluating
Debugging: Mechanics	Editing
Debugging: Stylistic	Revising

and planning the prose one wishes to write (Englert, Raphael, Anderson, Anthony, & Stevens, 1991).

Creating and Drafting

In coding and writing, both creating and drafting refer to the actual creation of the composition or project. The writer writes the words on the page, and the programmer uses the icons or text of the programming language.

Testing and Evaluating

In both coding and writing, testing and evaluating are the review process. The programmer watches her program run, and the writer reads her composition. This is a chance to see whether the composition accomplishes what the writer/programmer had planned. To do so, the writer must become the reader and the programmer must become the viewer and observe the results of the computer's compiler.

Debugging and Editing and Revising

Debugging in coding, like editing and revising in writing, is the activity of fixing one's existing composition. This can involve mechanical errors (editing); for example, the programmer forgot a start block in her programming composition and so the character did not move, or the writer started a sentence without a capitalized letter. It can also involve stylistic changes (revising). For example, the programmer changes her program to have a character move fewer steps, or the writer adds particular details to her written composition to help clarify. In both cases, the composer reviews her composition, identifies the gaps, and then engages the debugging/editing and revising process to address those gaps.

Compositional Activities in Dialogue

The fact that both compositional activities—writing and coding—involve these four subactivities, combined with the emerging reality that students are beginning to study coding at the very same time they are learning to write, provoked our curiosity about how the two learning processes might interact. We embarked on this research to explore students' engagement with these two compositional activities. What does the compositional act look like for emerging writers and programmers? What aspects of the process are young students more drawn to, and why?

Context of the Study

This ethnographic-interview study took place in one second-grade classroom at a private K–8 school in the Boston area. We chose this school because its literacy curriculum and coding curriculum are both robust. The writing curriculum has two parts: Fundations, focused on foundational skills (e.g., phonics and spelling), and writers' workshop, focused on narrative writing. The coding curriculum focused on the introductory programming language of ScratchJr, a block-based language designed for preliterate children. In block-based programming languages, programmers directly manipulate graphical elements called "blocks" instead of writing code using alphabetic characters. This interface introduces students to the concepts and practices of programming without needing to master the intricacies of more complicated programming languages (Weintrop & Wilensky, 2015). Almost all elementary school introductory coding classes use block-based languages.

We chose three focal students who represented a wide range of literacy levels in the class and who were eager to participate in the interviews. We began observing at the beginning of the school year and stayed until winter break. We watched the three focal students in their coding period and in their writers' workshop period and interviewed them on a regular basis. At first, we just spoke to them about their work in class and during choice time, the students' free block. As we got to know the three students better, we began (mid-November) conducting open-ended compositional interviews that were not connected to their work in class. These video and audio recorded interviews happened one or two times a week and cycled between writing and coding.

We then watched and transcribed all of the interviews. As we watched the open-ended compositional interviews, we looked for when the students engaged in the four subactivities of composition (planning/prewriting, creating/drafting, testing/evaluating, debugging/editing and revising) as well as any other activities they did while composing. What we found was that the students engaged all four subactivities while composing in writing and composing in coding, but the focus of their compositional activity was very different in each medium. Most significantly, while composing in coding, all three students debugged frequently. In contrast, while composing in writing, all three students were quite reluctant to revise and edit. To explore these findings, we introduce two of

the three students in the following sections. Each student represents one end of the spectrum in the class, as determined by the teacher's assessment of their reading and writing levels.

Sarah as Composer

Sarah's teacher described her this way: "Sarah is right on target. She's advanced while still being within the spectrum of the curriculum." In their phonics lessons, Sarah was in the highest level group. She hit the specific benchmarks for spelling, decoding, and basic grammar. Sarah enjoyed writing and wrote in a variety of genres. During writers' workshop, Sarah worked continuously on a "special moments" story that she composed over the four months of observation. This was a nonfiction piece about her first dance performance. During choice time, Sarah worked alongside her friends Amy and Lila on a collaborative writing project: a series of fictional picture books about Apple and Banana, two friends who have a series of adventures together. During their Fundations lessons, Sarah wrote journal entries on a range of topics from her ideal superpower to her dreams for adulthood to her weekend plans. Sarah was willing and happy to write, but while engaged in the activity of written composition, she focused mainly on planning and drafting. Very little time went into revising or editing. Specifically, when we divided all of Sarah's open-ended writing interviews into the various subactivities done in the process of writing, we found that editing and revising constituted only 6.6% of the activities in her process.

Sarah was also a very eager and excited programmer. During our open-ended programming interviews, Sarah developed a single programming composition that she worked on over the course of five interviews. Her programming composition was loosely based on her writing composition. By the beginning of her second programming interview, Sarah had an almost complete draft of her programming composition (see Figure 4).

The scene opens in a forest at a creek. A tiger (not a fox, despite the title) and a turtle are on screen. Tiger, through a voice recording, says to Turtle:

> "Hi, Turtle, do you want to play?"
> Turtle responds, through a voice recording:
> "Sure, Tiger, what game should we play?"
> Tiger responds:
> "I don't know, what game do you want to play?"
> Turtle responds:
> "Hmm, how about tag?"
> "OK, Turtle. I'll be it."
> "OK, give me a minute." Tiger then counts out loud to 60.

At this point in Sarah's program, the two protagonists, who had not yet moved, run around playing tag. 2

In her second and third open-ended programming interviews, Sarah tested her program. Her first problem was that the dialogue in her program did not work; she had mixed up the messaging blocks in the programming language when composing. Without hesitation, Sarah debugged this part of her program. Her second problem was that the voice recorder could not make a single recording that could fit a count to 60 (Sarah was recording herself, as Tiger, counting "1, 2, 3…" and putting the recording into her program). She debugged this part of her program as well, creating three different voice recordings to get her entire recording of Tiger counting to 60. Her third problem was in her program's sequence. Sarah wrote her program in such a way that the game of tag overlapped with Tiger's counting and was completed long before Tiger reached 60. Sarah had to rewrite her program using particular delay blocks in the programming language to ensure that the game of tag did not begin before Tiger finished counting. Sarah debugged with ease and eagerness (see Figure 5).

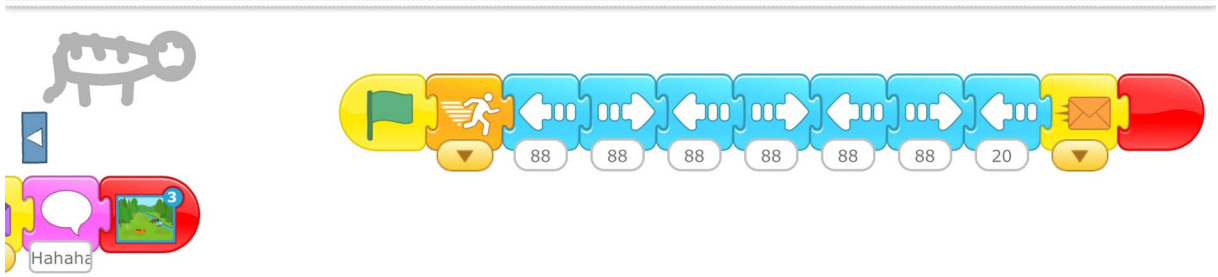
Sarah was also interested in fixing the aesthetics of her program. For example, after reviewing and evaluating her program, she decided that she did not want Tiger to run "into the stream," she said. She returned to her written program and debugged it to make sure that Tiger, when running around the screen during the game of tag, would always stop before the picture of the stream in the background. She also decided that she wanted Tiger to face a certain way, so she debugged her program to have him face that way. Later, she watched her program and evaluated that Tiger's movements did not perfectly reflect tag. She had written a program that had Tiger moving both up and down and side to side. A game of tag, Sarah explained, would only involve horizontal movements and not vertical movements. So, she returned to her program and debugged it so the program had Tiger moving only horizontally (see Figure 6).

Each of the last two examples, where tag should be played and what a game of tag looks like, were not issues with the "grammar" of the program (i.e., whether it was readable by the computer). Rather,

Figure 4
Sarah's Program

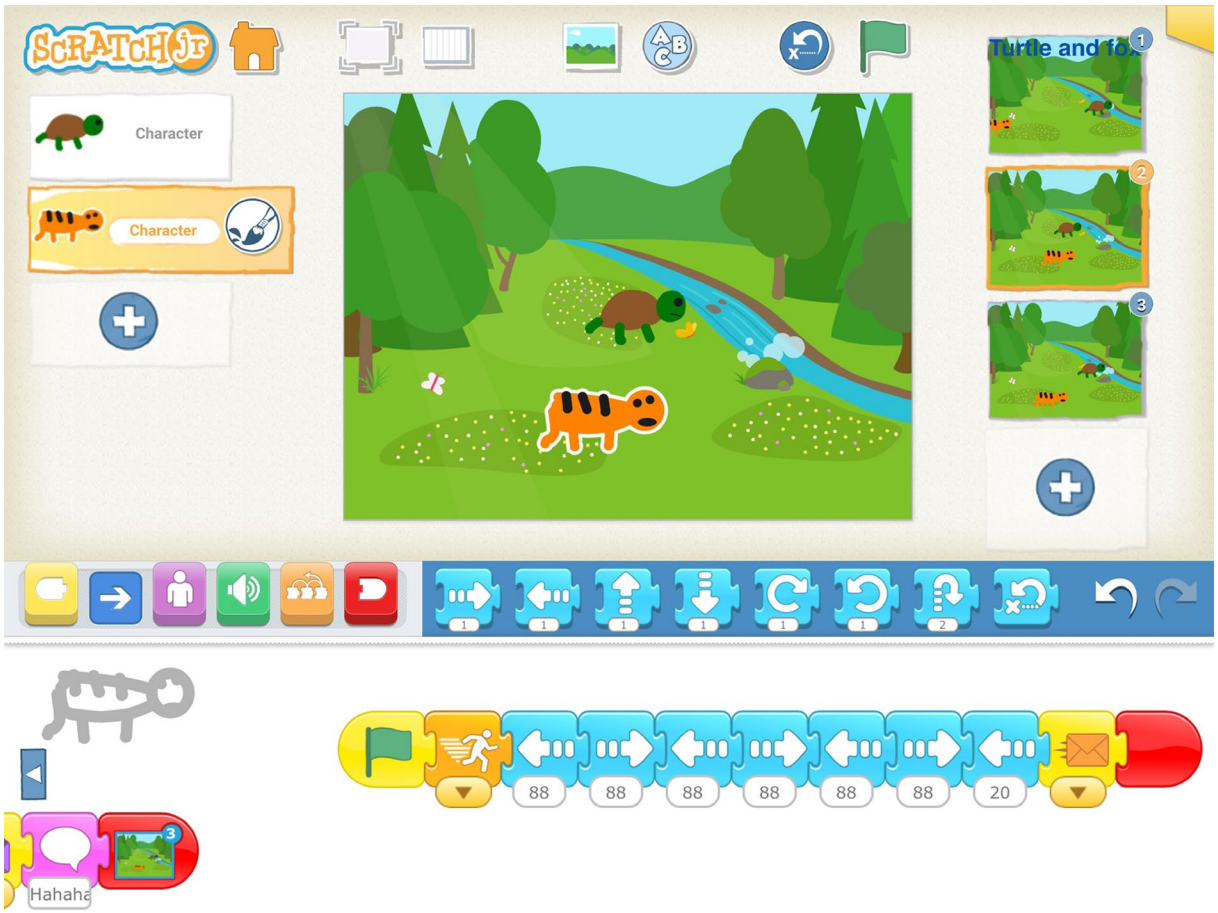
Note. The color figure can be viewed in the online version of this article at http://ila.onlinelibrary.wiley.com.

Figure 5
Sarah's Program's Mechanics Edited to Align With Her Plan



Note. The color figure can be viewed in the online version of this article at http://ila.onlinelibrary.wiley.com.

Figure 6
Sarah's Revised Program



Note. The color figure can be viewed in the online version of this article at http://ila.onlinelibrary.wiley.com.

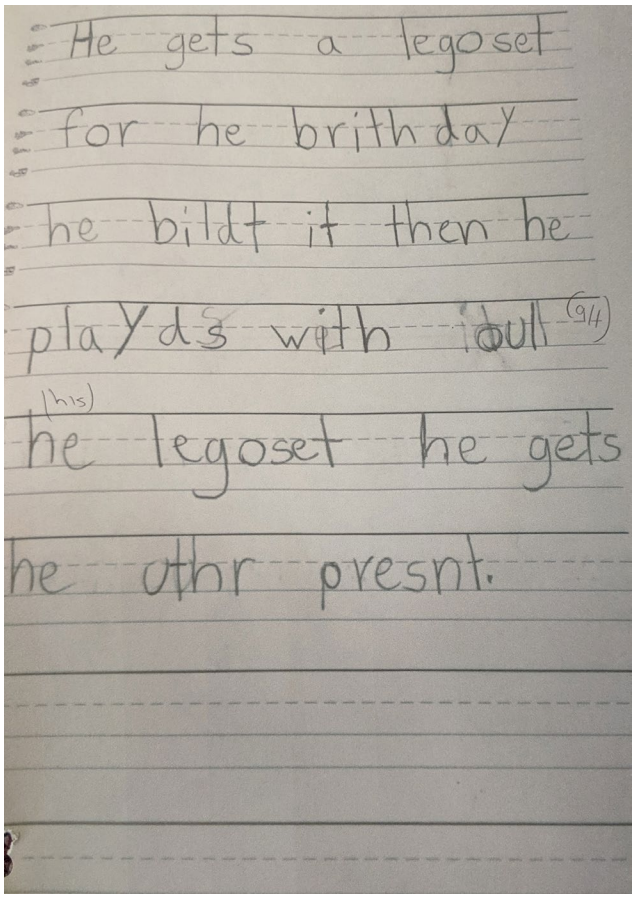
once Sarah had a program that ran, she attended to stylistic details in her composition and debugged her program to address these issues of style. In other words, Sarah was comfortable debugging both mechanical issues and stylistic issues. In fact, for Sarah, debugging constituted 20% of the activities in

her coding process, a marked contrast to her writing process.

Lila as Composer

Lila's engagement with writing and coding followed a similar pattern to Sarah's, but for Lila, more than for Sarah, the written compositions were born out of tremendous effort. The classroom teacher explained, "Lila is behind the grade-level benchmarks. She works very hard. I am not worried about her, but I am focused on her." Lila's first composition in the open-ended writing interviews, for example, was only 21 words long but took her 23 minutes to compose. Of those 21 words, Lila sought spelling guidance on 12. Her final product read (see Figure 7): "He gets a lego-set for he brithday he bildt it then he playds with oul (all) he (his) legoset he gets he othr presnt."

Figure 7
Lila's Final Written Composition



Note. The color figure can be viewed in the online version of this article at http://ila.onlinelibrary.wiley.com.

In composing this story, Lila reread her work three different times, but she did not make any edits or revisions. Besides the various mechanical mistakes (e.g., *he* vs. *his*, spelling of *all*, missing sequential words such as *then*, missing capitalization and punctuation), she also missed more substantive revisions that would have helped her reader understand the story. For example, who was the protagonist? What did he build with his Lego set, and what then was left over to play with?

It was not that Lila did not have a complete story in mind; she did. While composing, she explained that when her brother received the Lego set, he built it and then broke it down. Later, she explained that in the story she had in her mind, she comes into the scene and brings her own Legos, and she and her brother play together. However, after 23 minutes of writing, Lila was ready to stop. Upon stopping, she explained the rest of her story again, in more detail:

> My brother gets this Lego set. I'm jealous. I'm not really jealous, just in the story. So, I play with it a little bit and then steal it. And then my brother gets mad at me, so he takes the Legos back. We make up. I let him play with mine, and he lets me play with his.

All of this was in Lila's head—a creative nonfiction tale. In writing, however, she was only able to produce a fraction of it. In the process, she did no revising and no editing. In fact, in all of her open-ended writing interviews, Lila spent very little time revising or editing. Specifically, when we divided all of Lila's open-ended writing interviews into the various subactivities done in the process of writing, we found that editing and revising constituted only 4.8% of her process.

This was different than her programming compositions. There, in her first open-ended programming interview, she completed her first scene within minutes. Her program told the story of two protagonists, Apple and Banana, and how they met (the larger Apple and Banana saga is the storybook series that the three focal students worked on together in their choice block). In this origin story that Lila composed through coding, Apple goes to school for the first day and meets Banana for the first time.

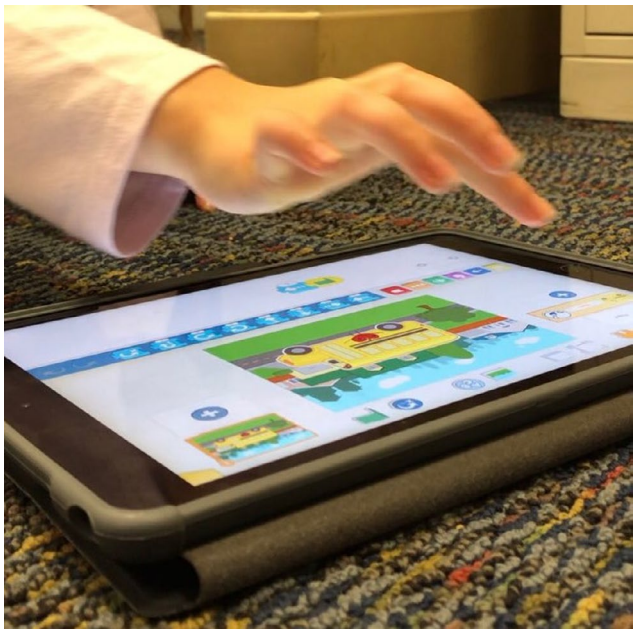
After drawing and selecting her characters and picking her background for the story, Lila announced, "OK, now let's start programming." Within minutes, she had completed the first part of her program (see Figure 8). She watched this first part of her program (a bus moving from one side of the screen to the

middle of the screen, where Apple was standing) and exclaimed in evaluation, "Yay! Just right!"

For the next part of her program, she wanted to get Apple to disappear and then for the bus to move off the screen so it appeared that Apple boarded the bus and it drove off to school (see Figure 9). The first step, Apple disappearing after the bus stopped, was challenging for Lila. She could not remember which block in the programming language caused a character to disappear. She tried a few different blocks, engaging the debugging process, and then figured out the right block. The second step, to get the bus to move again, was straightforward for Lila.

After programming these next parts, Lila reviewed her entire program. After rewatching, she evaluated it, deciding that she did not like that the bus went off-screen and then looped around onto the screen on the other side. So, without hesitation, Lila debugged her program, repeatedly changing the distance the bus was instructed to move until she succeeded in revising her program so the bus went entirely off-screen without reappearing. Lila's commitment to realizing her entire story as she wished it to be expressed in her program was in marked contrast to her reticence to revise and even edit in her writing composition. Like Sarah, Lila was able to engage in the revision process with her programming composition in a way she could not with her writing composition. Whereas revising and editing constituted only 4.8% of her activities in her writing process, debugging constituted 19% of Lila's activities in her coding process.

Figure 8
Lila Programming in ScratchJr



Note. The color figure can be viewed in the online version of this article at http://ila.onlinelibrary.wiley.com.

Figure 9
Lila's Debugged Program

Note. The color figure can be viewed in the online version of this article at http://ila.onlinelibrary.wiley.com.

Understanding the Emerging Composer

As emerging writers and computer programmers, Sarah and Lila engaged in the four subactivities of composition: planning/prewriting, creating/drafting, testing/evaluating, and debugging/editing and revising. Interestingly, however, their focus was consistently weighted differently when writing and when programming. When writing, they were eager to plan their composition and willing to convert their plan to the page, asking spelling and grammar questions as they went, but they were reluctant to edit and revise. This finding has been documented before in literacy research (see, e.g., Applebee et al., 1986; Fitzgerald & Markham, 1987). What is novel here is the juxtaposition to the way they composed in coding. When programming, Sarah and Lila allotted much more time to debugging.

What accounts for these differences in composition? Perhaps it simply comes down to the

affordances of each medium (screen vs. pen and paper). It is much easier to add, remove, and rearrange blocks on a screen than it is to erase pencil in a notebook. Or perhaps it reflects relative motivation and engagement, that the students find coding more fun, which in the interviews they made clear that they did. Although both of these reasons account for some of the difference, they might not account for all of it. In the vignette, Lila was able to watch her program (a bus moving from one side of the screen to the middle of the screen) and exclaim "Yay! Just right!" In her exclamation lies a key difference between composing in coding and writing.

Revision Process

Sarah and Lila's resistance to revision in writing, contrasted with their willingness to debug during coding, reflects a fundamental difference between the two compositional activities. At the heart of both the obstacle in writing and the affordance in coding are the slightly different roles of feedback and audience.

Feedback, Audience, and Purpose

The audience for the writer is abstract and external (Baker, 1994). Nystrand and Himley (1984) explained that although beginning writers intuitively grasp the give-and-take of an oral conversation, they have "little if any awareness of the idiosyncratic resources of written language" (p. 206). In speech, we are able to gauge the effectiveness of our message by the verbal and nonverbal responses of the addressee (Kress, 1994). This allows for much to be left unsaid. Writing, however, asks the author to achieve a rhetorical effect on a reader who is separated by time and space. This separation requires the author to strive for balance between her own expressive needs and the needs of the readers (Nystrand & Himley, 1984). To achieve this, the author must have a real sense of purpose for this more difficult form of expression.

All too frequently, the novice writer forgets or ignores the requirements of her reader and assumes that if she understands what is going on in the composition, so will her audience. In contrast, coding narrows the distance between composer and audience and offers emerging programmers more immediate and accessible feedback. In fact, there are two audiences: the computer and the user. The computer, as audience, is responsible solely for evaluating the mechanics of the composition (i.e., the "grammatical" correctness of the code). As audience, the computer gives the composer immediate and clear feedback on whether the grammar of the program worked. Either it runs or does not run. It would be equivalent to the pencil used by the writer piercing the page whenever a spelling or grammar mistake occurred.

This clear, immediate feedback from the computer makes space for stylistic evaluations and revisions aimed at the second audience, the user, which includes the composer and all potential viewers. The composer can immediately view her composition, which allows her to more easily understand how another audience will experience her composition.

Emerging Writers and Audience

The challenge of the new role of audience in written composition is evident in Lila's first composition. She begins her title-less story, "he gets a legoset for he birthday." Lila does not grasp that her audience has no idea who "he" is. In speech, she would not have to clarify this. Naturally, by the time one is telling a detailed story about Lego building and sibling rivalry, the audience would know that the speaker, standing right there, is speaking about her brother. One need not state it explicitly.

Something similar happens in Sarah's rewritten "Fox and Turtle" composition. Although Sarah had in her mind that Turtle's mother calls Turtle in for dinner, she did not attribute the quote "turtle time for diner" in her composition. Whereas this makes sense in speech, in which one can use intonation and expression to signal the switch from prose to dialogue and the interlocutor understands, Sarah may not yet grasp that her audience in writing has no way of knowing that this is dialogue.

Writing demands the development of "the habit of explicitness" (Kress, 1994, p. 36). Without the shared knowledge and reliance on the implicit that exists in speech, clear order and sufficient detail become essential features in written syntax for expressing meaning. The author must ask herself, What details are essential for my reader to understand my writing? How can I organize those details and narrative so the meaning I wish to convey is conveyed?

The challenge for the writer is that without direct and immediate feedback from her audience (i.e., given the less obviously iterative character of writing; Nystrand & Himley, 1984), gauging which details are necessary is not a simple matter. When the student writes and rereads her writing, she is filling in the gaps in her head all the time, without even realizing it. Of course, her audience cannot do that

because they do not have the context. Therefore, to understand her audience, she needs to take an imaginative leap outside her own thoughts. The writer must somehow play the roles of both author and audience. She must reread her story and ask herself, If I did not know what story I had in mind, would the words on this page adequately communicate it?

The affordance of coding composition is that when she is viewing the program, the student notices what is not clear on the screen, just as her audience would. In looking at the screen, she effortlessly transitions from composer to audience. We suggest that this easier identification with the audience may be at the root of why the students in this study had a greater willingness to revise their computer programs than their written compositions.

Implications for Teaching

We already know that reading and writing instruction can support one another. When students are taught to evaluate what they read (Duke & Pearson, 2002), they become increasingly aware of the options they have as writers. They can, as Hansen (1987) explained, mine the print for ideas for their own writing. They can evaluate, borrow, or revise authors' "ways with words."

What has yet to be considered seriously in literacy education is where coding and writing intersect. As two compositional activities students learn to do, they have much more in common than we have previously theorized (Vee, 2017). Both are students' initiation into composition, the activity of creating an artifact that communicates explicit meaning, meaning that can travel away from the author and even be understood entirely independent of the author. They both involve planning, writing, evaluating, and editing and revising.

We believe that when teachers recognize writing and coding as similar compositional activities, they can harness students' enthusiasm for revising in coding to support the development of their revision in writing. Perhaps, if students see that debugging in coding composition is the means by which clarity is achieved, they will be more willing to make the effort to engage in the revision process in writing. We suggest that language arts teachers create space in their curriculum for students to interpret or expand on stories they have written in the medium of code. They can then revise their written compositions in light of their coding compositions. For example, in an integrated curriculum we designed, focused on Maurice Sendak's *Where the Wild Things Are*, we ask students to write about what they would want to do in their own Wild Rumpus, then express that story in code, and then revise their written story (Hassenfeld, Govind, de Ruiter, & Bers, 2019).

This is not an argument about transfer but rather about the potential promise of teaching composition more broadly. Just as reading with the author in mind (Hansen, 1987) can bring into focus the choice points in writing, so too coding with the audience in mind might bring into focus the role of revision in writing. For the students in this study, writing composition and programming composition remained siloed. The next step is to bring them together explicitly, allowing students to compose the same narrative in both mediums and see if the enthusiasm students bring to the revision process in coding can be carried over to the revision process in writing.

TAKE ACTION!

We believe that one of the biggest challenges for novice writers is the imaginative leap required to engage in the revision process. The all-knowing author must transition to the unknowing reader:

1. Engage your students in other kinds of compositional activities that engage the revision process differently, such as coding.
2. Ask students to vocalize their thoughts as they revise and use their comments to identify challenges.

NOTE

The programming language that was taught in the classroom observed in this study was first developed in Marina Umaschi Bers's lab. This programming language is a free app used globally and is specifically designed for early elementary school grades. The findings of this research are in no way tied to this particular programming language.

REFERENCES

Applebee, A.N., Langer, J.A., & Mullis, I.V.S. (1986). *The writing report card: Writing achievement in American schools*. Princeton, NJ: National Assessment of Educational Progress, Educational Testing Service.

Baker, E.C. (1994). Writing and reading in a first-grade writers' workshop: A parent's perspective. *The Reading Teacher*, *47*(5), 372–377.

Duke, N.K., & Pearson, P.D. (2002). Effective practices for developing reading comprehension. In A.E. Farstrup & S.J. Samuels (Eds.), *What research has to say about reading instruction* (3rd ed., pp. 205–242). Newark, DE: International Reading Association.

Dyson, A.H. (1989). *Multiple worlds of child writers: Friends learning to write*. New York, NY: Teachers College Press.
Eckhoff, B. (1984). How reading affects children's writing. In J.M. Jenson (Ed.), *Composing and comprehending* (pp. 105–114). Urbana, IL: ERIC Clearinghouse on Reading and Communication Skills.
Englert, C.S., Raphael, T.E., Anderson, L.M., Anthony, H.M., & Stevens, D.D. (1991). Making strategies and self-talk visible: Writing instruction in regular and special education classrooms. *American Educational Research Journal*, *28*(2), 337–372. https://doi.org/10.3102/00028312028002337
Fitzgerald, J., & Markham, L.R. (1987). Teaching children about revision in writing. *Cognition and Instruction*, *4*(1), 3–24. https://doi.org/10.1207/s1532690xci0401_1
Gee, J.P. (2003). *What video games have to teach us about learning and literacy*. New York, NY: Palgrave Macmillan.
Graves, D., & Hansen, J. (1983). The author's chair. *Language Arts*, *60*(2), 176–183.
Hansen, J. (1987). *When writers read*. Portsmouth, NH: Heinemann.
Hassenfeld, Z., Govind, M., de Ruiter, L., & Bers, M. (2019). If you can program, you can write. Manuscript submitted for publication.
Kress, G. (1994). *Learning to write*. London, UK: Routledge.
Kress, G. (2003). *Literacy in the new media age*. London, UK: Routledge.
Kress, G. (2010). *Multimodality: A social semiotic approach to contemporary communication*. London, UK: Routledge.
Langer, J.A., & Flihan, S. (2000). Writing and reading relationships: Constructive tasks. In R. Indrisano & J.R. Squire (Eds.), *Perspectives on writing: Research, theory, and practice* (pp. 112–139). Newark, DE: International Reading Association.
Leu, D.J., Coiro, J., Castek, J., Hartman, D.K., Henry, L.A., & Reinking, D. (2008). Research on instruction and assessment in the new literacies of online reading comprehension. In C.C. Block & S.R. Parris (Eds.), *Comprehension instruction: Research-based best practices* (2nd ed., pp. 321–346). New York, NY: Guilford.
Leu, D.J., Jr., Kinzer, C.K., Coiro, J.L., & Cammack, D.W. (2004). Toward a theory of new literacies emerging from the internet and other information and communication technologies. In R.B. Ruddell & N.J. Unrau (Eds.), *Theoretical models and processes of reading* (5th ed., pp. 1570–1613). Newark, DE: International Reading Association.
Marsh, J. (2011). Young children's literacy practices in a virtual world: Establishing an online interaction order. *Reading Research Quarterly*, *46*(2), 101–118. https://doi.org/10.1598/RRQ.46.2.1
Nystrand, M., & Himley, M. (1984). Written text as social interaction. *Theory Into Practice*, *23*(3), 198–207. https://doi.org/10.1080/00405848409543114
Strawhacker, A., & Bers, M.U. (2015). "I want my robot to look for food": Comparing kindergartners' programming comprehension using tangible, graphic, and hybrid user interfaces. *International Journal of Technology and Design Education*, *25*(3), 293–319. https://doi.org/10.1007/s10798-014-9287-7
Vee, A. (2017). *Coding literacy: How computer programming is changing writing*. Cambridge, MA: MIT Press.
Weintrop, D., & Wilensky, U. (2015, June). In M.U. Bers & G. Revelle (Eds.), *To block or not to block, that is the question: Students' perceptions of blocks-based programming* (pp. 199–208). New York, NY: ACM.

MORE TO EXPLORE

- Vee, A. (2017). *Coding literacy: How computer programming is changing writing*. Cambridge, MA: MIT Press.
- Activities at ScratchJr: http://scratchjr.org/teach/activities (TK)
- Code.org CS Curricula: https://curriculum.code.org (TK)

3