Code and Tell: Assessing Young Children's Learning of Computational Thinking Using Peer Video Interviews with ScratchJr

Dylan J. Portelance
Tufts University
105 College Ave
Medford, MA 02155
dylan.portelance@tufts.edu

Marina Umaschi Bers
Tufts University
105 College Ave
Medford, MA 02155
marina.bers@tufts.edu

ABSTRACT
In this paper, we present a novel technique for assessing the learning of computational thinking in the early childhood classroom. Students in three second grade classrooms learned foundational computational thinking concepts using ScratchJr and applied what they learned to creating animated collages, stories, and games. They then conducted artifact-based video interviews with each other in pairs using their iPad cameras. As discussed in the results, this technique can show a broad range of what young children learn about computational thinking in classroom interventions using ScratchJr than more traditional assessment techniques. It simultaneously provides a developmentally appropriate educational activity (i.e. peer interviews) for early childhood classrooms.

General Terms
Design, Human Factors, Languages

Keywords
Computational thinking, Assessments, Artifact-based interviews, Early Childhood, Education, iPad, ScratchJr, Programming, Coding

1. INTRODUCTION
Since Jeannette Wing's seminal call for pre-college educators to engage students with computational thinking—a skill set at the intersection of problem solving, information processing, and system design [13]—the theoretical construct has become a crucial learning objective for students [1,8]. While computer science education researchers have recently contributed a significant amount of work to a growing knowledge base about teaching and learning computational thinking, studies do not often focus on learners in early childhood. Even children at the young age of four years old can learn foundational computational thinking concepts [2,3] and this kind of learning can support their literacy [9], mathematical [10], and socio-emotional development [10]. With these affordances in mind, there exists a need to create developmentally appropriate tools, pedagogy, and assessments for the learning of computational thinking during early childhood.

The ScratchJr iPad app was designed to provide young children (kindergarten through second grade) with developmentally appropriate means to learn computer programming concepts and skills while creating animated stories and games [5]. The ScratchJr programming language is based on the pre-existing educational programming language, Scratch, whose target users are children ages eight and above. Intending to retain the "low floor" and "high ceiling" of the original programming environment, the designers of ScratchJr kept features such as character and background creation, but changed the vertical orientation of the programming scripts to horizontal (to resemble the writing process), scaled the number of available programming blocks down, and designed other developmentally appropriate features [10].

One of the primary intended uses of the tool is for teaching and learning in the early childhood classroom [10]. In order to make ScratchJr viable for this purpose, there needs to be ways to determine what students are learning while they participate in tailored curricula and program their own ScratchJr projects. This paper seeks to begin addressing this.

In this paper, we present initial findings regarding an adapted use of artifact-based interviews in the early childhood classroom for the purpose of assessing what students have learned about computational thinking. The interviews are meant to supplement other traditional assessment techniques, such as design scenarios and project portfolio analyses [5]. The specific use of these interviews in a peer video recorded manner functions as a method for assessing student learning that may elicit different findings than traditional assessments or video-based analyses with videos not recorded by peers [7]. It simultaneously serves as an activity that is developmentally appropriate for early childhood classrooms that use ScratchJr on the iPad as a tool for learning computational thinking.

2. STUDY
The purpose of this study was to test the peer video interviewing technique with young children's ScratchJr projects as artifacts. In doing so, we sought to explore which computational thinking concepts the interviews could elicit and the videos could capture. At the same time, the study investigated the developmental appropriateness of the interviewing activity for use in early childhood classrooms learning computational thinking with ScratchJr.

2.1 Setting
This study was conducted in three second grade classrooms at a local public elementary school. The school was selected on the

IDC '15, June 21 - 25, 2015, Medford, MA, USA
 ISBN 978-1-4503-3590-4/15/06 $15.00
DOI: http://dx.doi.org/10.1145/2771839.2771894

basis of a preexisting relationship between the research team and the principal as well as the school's ownership of enough iPads to provide each student with their own during lessons.

2.2 Participants

Sixty-six total second grade students participated in the study. For the purpose of the paper three interviews conducted with three different students were selected as part of a case study to identify different kinds of computational thinking. During the curriculum enactment, each regular classroom teacher was present at all times in their respective classroom and, in almost all cases, two researchers were present serving as ScratchJr tutors. Although the researchers led the classroom activities during "ScratchJr time", regular classroom teachers were encouraged to make adjustments to ensure the curriculum and its enactment were an appropriate fit for their students and classroom culture. Additionally, there were sometimes other adults present in the classroom such as paraeducators working with specific students and student teachers unaffiliated with the research team.

2.3 Curriculum

The research team designed and taught a 13-day one-hour-per-lesson curriculum enactment of an adapted version of the ScratchJr "Animated Genres" curriculum[1]. This curriculum engages students in different challenges designed to build skills needed towards making final projects in the form of animated collages, interactive stories, and games.

2.3.1 Overview

The ScratchJr "Animated Genres" curriculum gives young children the opportunity to learn all the specific features in ScratchJr as well as "powerful ideas" from computational thinking such as debugging and modularization. These ideas can be usefully applied to many domains outside of computer science and engineering such as social science, writing, and music [11].

The curriculum was divided into three modules, each beginning with two lessons and ending with one day dedicated to working on an "animated genre" project and one day for participating in the interview activity. At the end of the study, we hosted an open house for families to come see what their children had learned and created throughout the curriculum enactment. Table 1 provides more details about the curriculum outline.

2.3.2 Projects

On Days 3, 7, and 11, students were given an hour to design and make a project within one of three different "animated genres."

2.3.2.1 Collage Project

For the collage project, students were instructed to fill a page with a combination of programmed characters. Leading up to the collage project, students learned how to use programming blocks to make their characters move around the screen and change the characters' looks. They also were taught how to create new characters by selecting default ones from the ScratchJr character library or designing their own using the Paint Editor. Finally, they learned how to start a programmed script using the "Green Flag" and color the background of their canvas.

2.3.2.2 Story Project

The story project allowed students to make use of text, sounds, and pages as well as control flow programming blocks like the "Repeat" block to create a narrative about their characters in ScratchJr with a beginning, middle, and end.

2.3.2.3 Game Project

Finally, the game project let students design user interactions with the "Start on Tap" block as well as use programming blocks that only work in certain scenarios like "Send Message," "Start on Message," and "Start on Bump." Students spent this hour creating a game with rules, an objective, and a control scheme for the player. They learned the most complex programming concepts in ScratchJr through game design.

Table 1. ScratchJr "Animated Genres" Curriculum Outline

Day	Module	Activity
1	Create a ScratchJr Collage	Learn about Motion blocks
2		Learn about Looks blocks, "Start on Green Flag" block, Characters, and Backgrounds
3		Make ScratchJr collage
4		Interviews with partners about ScratchJr collages
5	Create a ScratchJr Story	Learn about "Repeat" block, "Set speed" block, and "Wait" block
6		Learn about Text, Pages, Control blocks, Sound blocks, and End blocks
7		Make ScratchJr story
8		Interviews with partners about ScratchJr stories
9	Create a ScratchJr Game	Learn about "Send Message" block, and "Start on Message" block
10		Learn about "Start on Tap" block, and "Start on Bump" block
11		Make ScratchJr game
12		Interviews with partners about ScratchJr games
13	Family Day	Show family and friends ScratchJr collages, stories, and games

2.3.3 Interviews

Each day that followed one of the project days (days 4, 8, and 12), students met in pairs to conduct their peer artifact-based video interviews using their most recently created ScratchJr "animated genre" projects. Regular classroom teachers selected pairs of students based on a history of working well together in other activities. Pairings were retained throughout all three interviews.

The interviews were introduced to students as a way for them to show their ScratchJr projects to their classmates. Students were also informed that researchers would be viewing the interviews as a way to understand what students know about ScratchJr and programming so in the future they could better design the tool.

[1] http://www.scratchjr.org/teach.html#curricula.

During each interview, one student, the interviewer, would have her iPad camera app open on the video setting for filming and the other student, the presenter, would have her ScratchJr app open to show her project to the camera. On every interview day, each student participated as both interviewer and presenter.

Interviews comprised the following questions, designed by researchers, and displayed prominently on the classroom's whiteboard throughout each interview day.

- Tell me about your project.
- How did you make your project?
- What would you do if you had more time?
- Question of your choice

Students were told that the last question could be about anything they wanted to know about their partner's ScratchJr project.

3. RESULTS AND DISCUSSION

This paper presents the initial findings from the first study using peer artifact-based video interviews in an early childhood classroom using ScratchJr. These initial findings stem from analyses of three selected interview videos as well as field notes regarding the effectiveness of the activity.

3.1 Video Analysis

We present excerpts from three specific videos that demonstrate what kinds of computational thinking concepts can be elicited from the interview activity. Along with each excerpt, we provide a brief discussion about the concepts that the excerpt exemplifies. These excerpts were chosen based on the clarity of presentation.

3.1.1.1 Collage Project Interview

Excerpt

[1:52] Interviewer: How about because you're having trouble with this, [name redacted], why don't you, um, show us something you could make your character say? Why don't you make your character say something?

*[2:01] Presenter: Sure! I just did! *turning iPad around to face camera* I just made him say, "Hi [name redacted]." Look! "Hi [name redacted]."*

[2:09] I: That's great.

[2:10] P: Now, now I'm gonna make my dragon say, "Hi Bob."

[2:11] I: I like it.

[2:17] I: Okay.

[2:17] P: Would you like to see how I do it?

[2:19] I: Sure!

[2:21] P: Here, so, I just press the "Hi" button. "Hi." And now I click the, click the, "B," and now, "O," and then, "B." B-O-B, do that…, "Hi Bob."

[2:39] I: And then, what do you press now? Do you press, "Go"?

[2:40] P: "Hi Bob."

Analysis

In the excerpt above, the presenter, after careful deliberation and with the iPad facing away from the camera, delightfully shows the interviewer an example of how to use the "Say" programming block in a personally meaningful way. He then *reuses* the "Say" block in a new way to demonstrate his knowledge of what it does.

The video captures the student's knowledge and demonstration of a specific computational thinking practice, *reusing*, an important practice for efficiently choosing and returning to the right tools to perform a certain computational expression.

3.1.1.2 Story Project Interview

Excerpt

[3:19] Interviewer: What were you gonna put in for the tornado?

*[3:23] Presenter: Well, I would get the tornado to spin, and I would get that girl *camera does not show which girl the student is pointing to* to, um, be running away from the tornado and the tornado's following her.*

[3:32] I: And then eventually she would shrink, and then eventually you would make her disappear, like she got…

[3:38] P: Yeah, I would put her right…I would stop her, and then I would keep on getting the tornado to go, and the tornado would hit her, and I would make her disappear.

[3:50] P: I actually wanna do that. I think I'll do that right now.

Analysis

In this second excerpt, the presenter demonstrates how she could employ *sequencing* and *parallel programming* to express the next event in her story. While the student does not mention or point to specific programming blocks, each of the occurrences in the sequence that she proposes corresponds directly to one of the programming blocks available in ScratchJr. She mentions disappearing, moving, and stopping, all of which could be easily programmed using the shrink, hide, and motion blocks.

In this case, the video allows us to see the student's use of concrete computational thinking concepts in her speech without demonstrating those concepts in physical programs. Since computational thinking involves problem solving on both a human and computer level, this excerpt could begin to provide evidence of a computational thinking learning outcome for the presenter.

3.1.1.3 Game Project Interview

Excerpt

[2:12] Interviewer: If your iPad can only do three pages, what would you do?

*[2:19] Presenter: Well, I think I would make it so that when it…*taps the thumbnail for the third page to show the third level* I would make it so like, so half of this thing *makes a circling gesture around the page* when you reach, so, I would put these things *points to some of the characters* here, *points to left side of the screen* like some are here, and then when you *taps the thumbnail for the fourth page to show the fourth level* reach that part *taps the thumbnail for the third page to show the third level* then you would, um, when you reach this *points to the character that the game's hero needs to get to in order to advance to the fourth level* you go to the maze on the other part of this *points to the right side of the screen*, so on this half, so that's what I would do.*

Analysis

In this last excerpt, the presenter describes how she would solve a problem that the interviewer presents, a hypothetical lack of space on the screen. She immediately explains that if she were only given three pages of space instead of four, she would move everything from the fourth page to the right side of the third page. She would also squeeze everything previously on the third page to the left side of the page. Finally, she would program the game such that when the character finished the third level it would move to the page's right side rather than go to the fourth page.

The presenter exemplifies problem solving to address the concern of *space cost.* While space cost usually refers to space in memory, there are direct parallels between this student's "hacker-like" problem solving and that of a computer scientist trying to make the most out of limited memory. Dealing with constrained space is a problem within the scope of computational thinking and the video shows the presenter's ability to address it.

3.2 Best Practices for Classroom Use

Based on field notes taken by researchers during the enactments of the interview activity, there are several takeaways that could be drawn from to make the activity more effective in the future.

3.2.1. Do a practice run

In order to help students warm up to the interview activity, it may be helpful to have them do a trial run with ScratchJr projects they create during the first two lessons. This will allow them to get a feel for the activity but also allow researchers to see which students may need the most preparation for the their interviews.

3.2.2 Demonstrate good camerawork

Students often move their camera around while filming or zoom in on a partner's face instead of their project. In order to ensure that they capture their partner's ScratchJr project and any demonstrations they might do, show them what good camerawork looks like with another adult. Explain that the videos are about the ScratchJr projects and not their partner's nose or how fast their camera can move. Enforcing a time limit of two or three minutes can also prevent students who get restless easily from moving their cameras around towards the end of the interview.

3.2.3. Demonstrate good presenting

While the iPad cameras are fairly good at picking up sound if the presenter is up close, it is up to the presenter to actually show ScratchJr to the camera. Just as good camerawork should be demonstrated by a pair of researchers, so should good presenting. Explain how the focus on the ScratchJr projects applies to both the interviewer and the presenter. Emphasize that the only thing that will show up in the video is what is facing the camera. If students are having trouble, it can be helpful to have the interviewer stand over the presenter while the presenter sits with their iPad on the table. This way both students can look directly at the iPad with the ScratchJr projects on it.

4. CONCLUSION

In this paper, we presented a way to use peer artifact-based video interviews in order to assess the learning of computational thinking in early childhood classrooms. As shown in the above discussion, students can showcase a broad range of computational thinking concepts as they demonstrate and talk about their personally meaningful ScratchJr projects for the camera. While individual clinical-style interviews with young children to construct case studies [12] have long been demonstrated as a productive method to understand children's thinking about their own learning of programming, conducting these interviews in a classroom environment, as a part of a computer programming curriculum, is logistically challenging. The artifact-based video interviews in pairs provide an alternative that seems to capture rich data aimed at contextualizing projects and understanding children's thinking. That said, this study is an exploratory one. Future research could identify if patterns appear in a larger corpus of students and if there are differences in how students across different demographics participate in this activity. Given the findings, refinement of peer artifact-based video interviews as a classroom activity and assessment method is worth pursuing for computer science education researchers focusing on young children.

5. ACKNOWLEDGMENTS

This project is supported by the National Science Foundation grant DRL-1118664.

6. REFERENCES

[1] Barr, V. & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54.

[2] Bers, M.U. (2008). *Blocks to Robots: Learning With Technology in the Early Childhood Classroom*. Teachers College Press, New York.

[3] Bers, M.U. (2012). *Designing digital experiences for positive youth development: From playpen to playground.* Oxford, Cary, NC.

[4] Brennan, K. & Resnick, M. (2012). New Frameworks for Studying and Assessing the Development of Computational Thinking. Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada.

[5] Clements, D.H. (1999). The future of educational computing research: The case of computer programming. *Information Technology in Childhood Education Annual*, 1999(1), 147-179.

[6] Flannery, L.P., Kazakoff, E.R., Bontá, P., Silverman, B., Bers, M.U., & Resnick, M. (2013). Designing ScratchJr: Support for Early Childhood Learning Through Computer Programming. In Proceedings of the 12th International Conference on Interaction Design and Children (IDC '13). ACM, New York NY, USA, 1-10.

[7] Jordan, B., & Henderon, A. (2009). Interaction Analysis: Foundations and Practice. *Journal of the Learning Sciences*, 4(1), 39-103.

[8] Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2007). Computational Thinking for Youth in Practice. *ACM Inroads*, 2(1), 32-37.

[9] Kazakoff, E. & Bers, M.U. (2012). Programming in a robotics context in the kindergarten classroom: The impact on sequencing skills. *Journal of Educational Multimedia and Hypermedia*, 21(4), 371-391.

[10] Kazkoff, E., Sullivan, A. & Bers, M.U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal,* 41(4), 245-255.

[11] Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas.* Basic Books, Inc., New York, NY, USA.

[12] Turkle, S. (1984). *The Second Self: Computers and the Human Spirit.* Simon and Schuster, New York, NY, USA.

[13] Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.