From Spoken Words to Prompt Triggers: Technical Iterations of a Semi-Intelligent Conversational Agent to Promote Early Literacy

Brandon Hanks[1](✉), Grace C. Lin[1], Ilana Schoenfeld[1], and Vishesh Kumar[2]

[1] Massachusetts Institute of Technology, Cambridge, MA 02139, USA
{bhanks,gcl,ilanasch}@mit.edu
[2] Northwestern University, Evanston, IL 60201, USA
vishesh.kumar@northwestern.edu

Abstract. AI technology is rapidly evolving and has vast potential for educational applications. This paper focuses on the technical iterations that took place as our project team developed a semi-intelligent conversational agent (CA) that uses speech recognition to fire spoken prompts to promote caregiver-child interaction as they read books aloud together. Situating this work in a design-based research methodology, the technical iterations reported here are part of the iterative "build" phase. (Easterday et al., 2018; Hoadley & Campos, 2022). The CA app promotes conversations between caregivers and children by listening to the human dyads as they read, matching their spoken words to marker words that would pinpoint the page of the storybook the dyads are reading, and playing a prompt corresponding to the page. The dynamic system that supports the app involves multiple components: web accessible services, data processing services, and human outputs, and it has gone through a combined seven iterations in three prototypes. Though a very small part of the DBR cycle, the technical iterations presented here have the potential to inform others interested in incorporating text-to-speech and other AI technologies into educational applications. We close with considerations for future directions.

Keywords: Educational Technology · Early Literacy Software · Conversational Agent

1 Introduction

Decades of research have shown that back-and-forth conversations between more advanced readers, such as the parents, and novice readers, such as children, are essential for children's early literacy development [1–3]. These conversations can take place as caregivers and their children shop at the grocery store [4, 5] or as they read books aloud together. In fact, strategies have been developed to help caregivers engage children in dialogic reading practices [6–8]. For example, as caregivers read a book to their child, they may pause and ask children questions related to what they have been reading. These questions might include asking the child what they see on the page (e.g., "Do you seen an elephant?"), explaining vocabulary words (e.g., "What does the word vegetarian

P. Zaphiris et al. (Eds.): HCII 2023, LNCS 14060, pp. 58–69, 2023.
https://doi.org/10.1007/978-3-031-48060-7_5

mean?"), or relating to something in their everyday life to a picture they are seeing in the book (e.g., "What did you do the last time you went to the playground?").

This backdrop of dialogic reading practices is the inspiration behind our educational technology app. In this paper, we will highlight how we sought to promote dialogic reading practices through innovative technologies. Specifically, we will begin by establishing the methodology with which we carried out our work. We will then describe how the app works from both the user's perspective and, to get a sense of the dynamic system, from the perspective of a datum. We will also highlight the various challenges that surfaced in our rounds of technical iterations. Finally, we will conclude with the future directions.

2 Design-Based Research

Adhering to the advice of Reeves [9] that educational technology researchers should make the goals more explicit, our work presented in this paper focuses on the development goals and follows the iterations of development research. Specifically, the design and development of the dialogic reading app is grounded in design-based research (DBR; [10, 11]). DBR is a methodology used by educational researchers, designers, and developers to design and test out innovative approaches or technologies to improve learning. The process of DBR often involves multiple cycles and iterations, often beginning and ending with design principles and values [9–11]. After identifying our design values and providing a general overview of how the app is supposed to function, we highlight the iterations during the "build and test" phases of the DBR cycle.

2.1 Objectives and Design Values

Once "dialogic reading practices" had been identified as a goal, the team went on to establish design principles/values for the app. In particular, the team decided that the app must be:

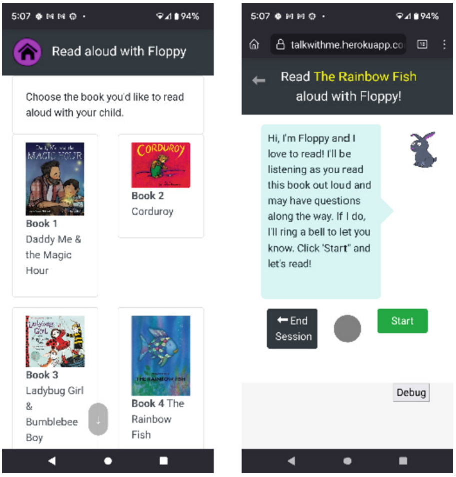
- Easy to use
- Neither a distraction nor the centerpiece of the experience
- Accessible

In particular, the concept is to have the technology scaffold caregivers who otherwise may not be following best-practices of dialogic reading while reading with their children. It should generate dialogue between the pair, and should neither be a distraction nor the centerpiece of the experience. We envisioned caregivers and children reading a physical book together while a conversational agent (CA) listened and guided the conversations they were having. The human dyads' conversations are the focal point the app serves to promote. Therefore, unlike eBooks where the human dyads had to directly interact with the technology (e.g., touch the screen to move on to the next page), we did not want the caregivers or the children to stare at a screen/device. Because of the notably lower language and literacy levels of students from socioeconomically disadvantaged backgrounds [12], we also settled on targeting families with lower annual household income. With this in mind, the technology must be accessible to most people (i.e., no AR or VR that requires a high-end phone).

3 The App: How It Works

3.1 Overview of the User's Experience

To use the app, the caregiver (or the child) first chooses a physical book they want to read (see Fig. 1). They select the book on the app and start the session. Once the "Start" button is pressed, the CA starts listening in. When they read to a page with a preset prompt, the device will play a chime sound alerting the human dyads that the CA is about to speak. Following the chime, the CA will fire the audio prompt. These prompts are meant as conversation starters between the caregiver and the child (i.e., the CA is not meant to engage in a conversation with the dyads). The prompts also serve to model for the caregivers the types of dialogic questions they could ask. For example, as a caregiver reads page 4 of Corduroy where the writer describes how Corduroy was sad as Lisa (a child in the story) and her mom walked away, a chime will sound followed by the CA asking, "Why did Corduroy feel sad?" The human readers will then talk about what it means to feel sad, why Corduroy was sad, etc., before they resume reading the book.



Choose the book you'd like to read aloud with your child.

Hi, I'm Floppy and I love to read! I'll be listening as you read this book out loud and may have questions along the way. If I do, I'll ring a bell to let you know. Click 'Start" and let's read!

Fig. 1. Screenshots of the app. On the left, we see the book selection screen. On the right, we see the screen (Prototype 3 Iteration 7) once the participant selected *The Rainbow Fish*.

3.2 Architecture

For the CA to work, a dynamic system was designed to connect the various moving pieces (see Fig. 2). The system involves multiple components: web accessible services (represented by ovals in Fig. 2), data processing services (rectangles in Fig. 2), and human outputs (diamond in Fig. 2). An essential feature of the system, marker words were created using n-grams: The books' texts are first sent to the backend, where a script written by our developer automatically determines the types (e.g. bi-grams, tri-grams) and density of n-grams on each of the books' pages. N-grams unique to individual pages are then identified as potential marker words. Content developers wrote dialogic questions for the pages of the books containing unique marker words and then sent both text and audio-recordings of the questions to the backend. These questions were then ready to be played aloud any time a user's recorded spoken words matched a recognized marker word.

Let's track the journey of a snippet of voice data to understand the system's connections better. When a caregiver and a child want to read with the app, the caregiver opens the app, grants microphone access, selects a book, and starts reading out loud. The voice data snippet originates when the caregiver presses the "start" button on the app, and leaves the front end immediately. When it reaches the backend, the snippet is appended to a short mp3 file that then travels to Google and is transcribed into text. The text returns to the backend, which converts it into an n-gram and tries to match it against n-gram marker words. If a matching n-gram marker word is found, the prepared audio prompt from the content developers is triggered and sent to the user-facing frontend, where it is played on the caregiver's device. Seemingly simple, this entire process continues until the reader decides to pause or end the read-aloud session. A final copy of the entire mp3 audio file is sent to a secure archive. Along the way, metadata are created to indicate session information, e.g. the time the caregiver accessed the app, the book selected, and other de-identified details about the user. These metadata are also sent via the backend to the secure archive.

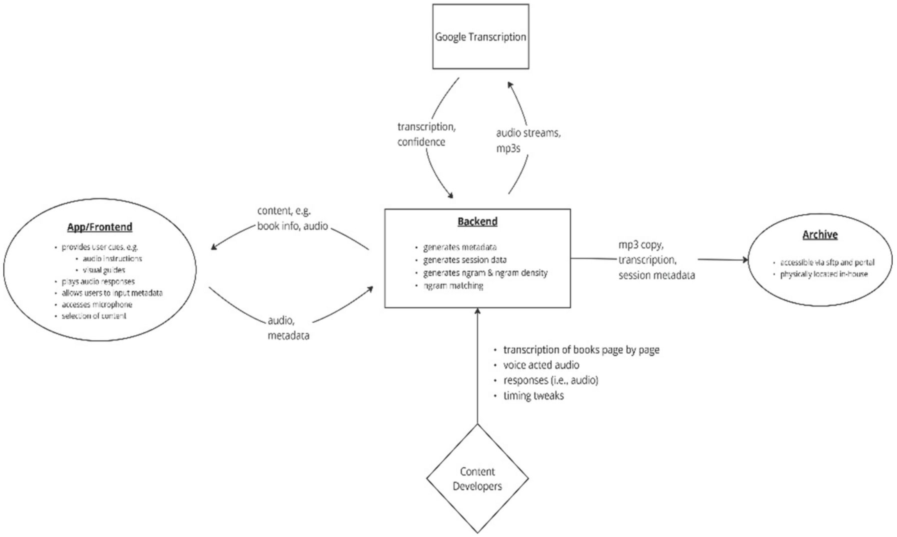

Google Transcription
transcription, confidence
audio streams, mp3s
App/Frontend
• provides user cues. e.g.
• audio instructions
• visual guides
• plays audio responses
• allows users to input metadata
• accesses microphone
• selection of content
content, e.g. book info, audio
audio, metadata
Backend
• generates metadata
• generates session data
• generates ngram & ngram density
• ngram matching
mp3 copy, transcription, session metadata
Archive
• accessible via sftp and portal
• physically located in-house
• transcription of books page by page
• voice acted audio
• responses (i.e., audio)
• timing tweaks
Content Developers

Fig. 2. Interconnected system for the conversational agent app.

4 The App: Technical Development Iterations

The system was not built all at once. Three prototypes that went through a combined seven iterations were created prior to testing the app in usability and efficacy studies. The first prototype, for example, used a robotic voice for the dialogic reading questions, but had all of the essential functions. The second prototype iterated on the user interface, swapped the robotic voice for a human recorded voice, and addressed feedback from educational and literacy experts. The third and final prototype was the one deployed for testing. Following, we detail the technical iterations from the Proof of Concept to the final prototype (see Table 1 for an overview).

4.1 Proof of Concept

The initial proof-of-concept was a Python script that used PyAudio [13] to capture audio, Google Cloud to transcribe it, and a system call to mpg321 [14] to play MP3 files. A second Python script was written to identify unique words in a body of text which could be used to determine when to play the MP3 files. That second script would then associate these unique words, aka "markers," with book pages and associated MP3 files. Next, some sentences were written as filler content, and converted to MP3 files via Google Text-To-Speech. Finally, the second script was run against a test book, The Very Hungry Caterpillar, which had been transcribed to a text file. We immediately discovered that the book contained very few unique words, and so to increase the likelihood of having more markers we switched to The Snowy Day and added unique word pairs, aka "bigrams." We then successfully demonstrated that the entire process worked both in concept and practice.

4.2 Prototype 1: Scalability, UI, Robotic Voice

Next, to both increase the scalability of the process, compatibility across platforms, and minimize disk use (which is at a premium on low-end devices), we decided to migrate the system onto the web. (See Table 1, Iterations 1 ~ 3.) For a back-end, we used Django with Channels for three reasons. First, since the proof-of-concept was written in Python, it allowed for simple porting of the demo codebase. Second, while streaming over the web is typically done via something like RTMP, constraining all of the data and messaging to a websocket allowed less system complexity. This allowed our tiny team of one developer, two designers, and two researchers to minimize maintaining web infrastructure. Third, Django is full- featured, allowing for development to be focused on the interfaces and data, rather than on the nuts-and-bolts of webservers. PyAudio and mpg321 were also both replaced by a front-end vanilla JavaScript app which captured audio via the browser's getUserMedia [15] method and played audio by loading MP3s into HTML audio elements. The MP3 responses were generated algorithmically from written text which had been sent to Google Text-To-Speech for voice synthesis, and stored in the back-end for transmission over the websocket. These responses consequently sounded robotic, due to their synthetic nature.

We then collaboratively generated user stories for three groups of users: reading dyads, researchers, and content experts. Web apps were then written to support the three

Table 1. Technical Iteration Descriptions. Each major prototype is separated by a line. After the Proof of Concept, Prototype 1 went through iterations 1 ~ 3, Prototype 2 went through iterations 4 ~ 6, and Prototype 3 is currently on the 8th iteration.

Iteration #	Description	Feedback or Issues	Revisions
0	Proof of Concept	Not scalable	Moved to web architecture
1	Robotic voice with simple questions in browser	Needed more UI	Added more dyad UI, such as ability to pause recording and animated character
2	More dyad UI	Not enough books Prompts were not triggered well Needed better artifact capturing	Added interface to easily add more content Better n-gram matching and filtering
3	Researcher/Content Creator interfaces	Dyad UX issues such as how interruptive the voice was	Added chime, more indicators (markers), and debug messages for the dyad
4	Dyad UX enhancements	Questions were too simple or too complicated (depending on age of the kid) Timing of prompts not great	Created new questions Implemented new timing mechanism
5	Human-recorded voice	Timing issues remain	Added another timing mechanism Experimented with new books with more markers per page
6	Content creator timing tweaks	Audio not playing correctly in certain browsers	Migrated to standardized Android devices
7	Android devices/Usability Study	Requested repeat button Some prompts fired at the wrong place	Android App Repeat functionality added Revised n-gram lists
8	Android App/Efficacy Study	Data analysis is currently underway	Stay tuned

use cases, and an SFTP research repository was created to store session artifacts (such as MP3s) from the system. While a monolithic app, such as a Single Page Application (SPA), would likely have made the system feel more cohesive, we chose this approach both to separate the roles further and make it easier to iterate on features specific to those roles. For example, adding blinking lights to the dyad interface could be done without touching any of the other user groups' interfaces.

4.3 Prototype 2: UX Enhancement, Human Voice, and Tweaks

We completed an initial user test following the first prototype. Based on feedback, we enhanced the user experience by polishing the questions asked and swapping out the robot voice with human voice. We also uncovered technical problems with prompt timing and frequency during this initial test (see Table 1, Iterations 4–5). To address these issues, we experimented with two alternative mechanisms for transcribing the audio. The original mechanism involved streaming the audio directly from the back-end to Google, which would return a transcript upon speakers' natural pauses. The new second mechanism had Google return transcriptions immediately, though it had similar issues since it often contained errors and low confidences. The new third mechanism would create MP3s from the audio stream on a fixed interval of 8 to 13 s for submission to Google, which seemed to improve both the frequency and reliability of responses. A toggle was created to select which mechanism was to be used per book, and we tested each to determine which worked best. For most books, mechanism three with intervals of 8, 9, or 10 s worked well, though the original mechanism was retained for two of the test books.

Our subsequent user testing revealed yet another technical obstacle (Table 1, Iteration 6). At this point, the app had been solely an in-browser app to minimize hard-drive space. However, this benefit was not able to be realized since we found that some Chrome browser versions were preventing the audio files from playing successfully. Specifically, the anti-spam measures built into the browsers were the culprit. To solve the issue with media blocking, we moved the front-end into our standalone coaching app, which was built in Expo/React for Android devices.

4.4 Prototype 3: Android App and Continued Iterations

The third prototype was then created as an Android app to address these compatibility issues. Due to our values of accessibility and the belief that the technology should work even on low-end devices, we acquired older versions of Android phones for use in our first formal usability test. We standardized each device, ensuring that each phone had identical installs, operating systems, and permissions set. Since we were intending to deploy only into browsers at this time, this allowed us to minimize issues that might arise that were not related to our technology. It also allowed us to focus our attentions specifically on the technology, and not on attempting to troubleshoot with a distant user. To be specific, once the user clicked on a book in the app (see Fig. 1 left), they were taken to the browser-based page (see Fig. 2 right) where they could then start their reading session.

After the installation, we mailed the phones (with return envelopes) to the 20 participating families (18 mothers, 2 fathers; 10 boys and 10 girls) in our formal usability study. The dyads read with and without the app in remote Zoom sessions as well as at-home reading sessions. During the first Zoom session, participants first read without using the app. We then introduced them to the app and dialogic reading strategies. They then read again with the app. After providing feedback to the team, we asked the participants to read at home using the app before their next session scheduled 2–4 weeks later. The second Zoom session repeated largely the same procedure, though families no longer needed the app introduction.

After gaining the extended exposure to the app, the families revealed that the app was easy to use (see [16] for a more detailed report of the usability results). However, they also echoed our findings from previous iterations, such that the timing of the conversational agent could be off. Some participants even thought that it was because of their own reading pace. Because they occasionally missed a question prompt, multiple participants asked for a repeat button on the app.

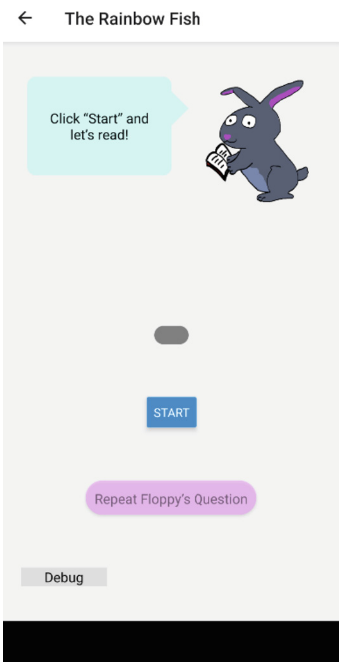
We built the repeat function into the newest iteration of the app (see Fig. 3) and addressed the timing concern this time by manually checking the auto-generated unique n-grams and removing the ones that were not so unique after all (e.g., “shimmer” was a marker word on page 3 of The Rainbow Fish while “shimmery” was a marker word for page 18).

In playtesting the new set of marker words, we also discovered a rarely occurring bug in which two sessions could cross-communicate triggers, due to how the temporary MP3s that were being sent to Google Cloud were stored. For example, one dyad could be reading a book at the same time as another dyad, which might trigger incorrect prompts to be activated in the second dyad’s session. Similarly, dyad 2’s reading could trigger the incorrect prompts for dyad 1. While we did not observe this happening in any studies, it may have been responsible for some of these early timing issues that we encountered during in-house testing. To ensure future users do not encounter prompts from the wrong books, we resolved the issue by implementing additional codes to prevent collisions among the temporary MP3s.

Additionally, because, among other things, we wanted to investigate how the app could function on participants’ own devices, this newest version is completely contained within the app (see Fig. 3); participants are no longer taken to a browser page to start their reading session. The newest version had now been implemented in an efficacy study on whether the educational app could help caregivers improve their dialogic reading practices. Data analysis from the efficacy study is still underway, though the analyses from the usability participants’ reading sessions with the app suggest the conversational agent indeed promoted conversations between caregivers and children [17].

5 Discussion

One persistent issue that gave us a lot of trouble was the timing of the responses by the agent. This is a commonly recognized challenge in the design and creation of conversational agents [18, 19], especially balancing the tension between processing speech inputs, deciding on appropriate responses, in a way that feels natural to the ongoing conversation. Corroborating similar work, we also experienced difficulty triggering prompts from the CA fast enough and with enough frequency. For the former, parents would sometimes have turned the page and would need to return to the previous one to have the discussion with the child. For the latter, if the agent was not responding often enough the parents would assume a technical issue had occurred and would turn their attention to the screen. This was exacerbated by our decision to ensure the technology would be



The Rainbow Fish

Click "Start" and let's read!

Fig. 3. Screenshot of the newest version of the literacy app. The "Repeat" button is displayed on the bottom, and participants are no longer taken to a browser page while reading.

deployable on low-end devices. As a result, given our resource constraints we decided to switch to an Android app rather than build out a technology stack that could respond with increased optimal timing. As recommended by other conversational agent design work, possible solutions include developing backchannels for feedback so that users are aware of the agent's intent to participate in a more conversationally natural manner, though this also brings up a tension with our design where we did not want the conversation to be conducted with our agent itself but only augmented with one way prompts [18, 19].

On a positive note, the simple design of the tech stack made it possible for such a small team to make a tool that ultimately families enjoyed using. Using just websockets made it straightforward to switch to an Android app. It also made it easy to add and modify both content and how that content was used by the front-end, since no API endpoints needed to be configured to support the change.

Additionally, the decision to use the low-end devices in the first formal test with remote users turned out well. First, we were able to ensure that the technology worked even with phones that were not that fancy. Second, by standardizing each device and the installation, we were able to concentrate on the usability of our technology. For example, browser plug-ins that change website functionality, system permissions incorrectly set to block microphone access, or having the audio volume turned off could have prevented the technology from working. Although the setup and mailing process was time-consuming, it helped us avoid encountering issues that may be irrelevant to the application we were developing.

Some lingering issues remain with regard to our literacy app's content. We tested 15 books throughout development, and only eight seemed to have enough unique markers to allow prompts frequent enough to engage the parents. Children's books often contain repetitions to help children get into the rhythm of reading and reinforce learning concepts [20]. This pattern makes it challenging at times to identify books that work well with the app as they must contain unique marker words to trigger the types of dialogic reading prompts we wanted to target. There is also a copyright-related issue related to scaling the app for public usage. The app requires the uploading of the full, exact text of each of the books to be read aloud. Copyright permissions for popular children's books are often difficult and/or costly to obtain. Books also go out of print at times, making it harder to obtain hard copies.

To address aforementioned issues, one potential solution could involve the creation of a dedicated series of children's books designed to seamlessly integrate with the application. Furthermore, future endeavors could leverage recent advancements in both artificially generated art and conversation. For example, the current version of the conversational agent relied on prewritten prompts. Therefore, a promising avenue for future research could involve exploring whether new and different dialogic prompts that are generated each time the same book is read aloud might maintain a higher level of novelty and engagement in the caregiver-child dyad's reading experience over time (as it is common for young children to want to read the same story over and over again).

6 Conclusion

We developed a semi-intelligent conversational agent who asked questions related to children's books that caregiver-child dyads read aloud. The questions were meant to both model the dialogic reading strategies for caregivers as well as prompt conversations between the human dyads. Throughout development and multiple iterations of playtesting and formal testing, we uncovered various kinks and attempted to address them while adhering to our design principles/values of having the technology be accessible, easy to ease, and not the centerpiece of the experience. As we continue our exploration, we hope this technology can do its part in promoting early literacy, and that this paper can be helpful to others pursuing similar lines of work.

Acknowledgment. This work was funded by the Chan Zuckerberg Foundation as part of the Reach Every Reader project. We would like to thank all the playtesters in and out of The Education Arcade who gave valuation feedback throughout the development process. We would also like to give special shout-outs to Louisa Rosenheck, who led the project early on and established the design values and principles, Scot Osterweil, who designed and created the character that embodied the conversational agent, Melissa Callaghan, who, besides being an exceptional researcher and designer, also voiced the first iteration (human version) of the conversational agent, and Dr. Cigdem Uz-Bilgin, without whom the formal usability testing would likely have fallen apart. Finally, the work would not have been possible without Dr. Kathryn Leech, whose dialogic reading strategies are the basis of our early literacy app, and Dr. James Kim, who provided invaluable insights toward the prompts the conversational agent should ask.

Author Contributions. Brandon Hanks developed the literacy app and performed writing – original draft and conceptualization of this paper. Grace Lin performed writing – original draft and

conceptualization of this paper. Ilana Schoenfeld and Vishesh Kumar performed writing – review and editing.

References

1. Towson, J.A., Fettig, A., Fleury, V.P., Abarca, D.L.: Dialogic reading in early childhood settings: a summary of the evidence base. Top. Early Childhood Spec. Educ. **37**(3), 132–146 (2017). https://doi.org/10.1177/0271121417724875
2. Arnold, D.S., Whitehurst, G.J.: Accelerating language development through picture book reading: a summary of dialogic reading and its effect. In: Dickinson, D.K. (ed.) Bridges to literacy: Children, families, and schools, pp. 103–128. Blackwell Publishing, Malden (1994)
3. Doyle, B.G., Bramwell, W.: Promoting emergent literacy and social-emotional learning through dialogic reading. Read. Teach. **59**(6), 554–564 (2006). https://doi.org/10.1598/RT.59.6.5
4. Hirsh-Pasek, K., Golinkoff, R.M.: Put your data to use: entering the real world of children and families. Perspect. Psychol. Sci. **14**(1), 37–42 (2019). https://doi.org/10.1177/1745691618815161
5. Bustamante, A.S., et al.: More than just a game: transforming social inter-action and STEM play with Parkopolis. Dev. Psychol. **56**(6), 1041–1056 (2020). https://doi.org/10.1037/dev0000923
6. Leech, K.A., Rowe, M.L.: An intervention to increase conversational turns between parents and young children. J. Child Lang. **48**(2), 399–412 (2021). https://doi.org/10.1017/S0305000920000252
7. Leech, K.A., Wei, R., Harring, J.R., Rowe, M.L.: A brief parent-focused intervention to improve pre-schoolers' conversational skills and school readiness. Dev. Psychol. **54**(1), 15–28 (2008). https://doi.org/10.1037/dev0000411
8. Leech, K.A., Haber, A.S., Jalkh, Y., Corriveau, K.H.: Embedding scientific explanations into storybooks impacts children's scientific discourse and learning. Front. Psychol. **11**, 1016 (2020). https://doi.org/10.3389/fpsyg.2020.01016
9. Reeves, T.C.: Socially responsible educational technology research. Educ. Technol. **40**(6), 19–28 (2000)
10. Anderson, T., Shattuck, J.: Design-based research: a decade of progress in education research? Educ. Res. **41**(1), 16–25 (2012). https://doi.org/10.3102/0013189X11428813
11. Barab, S., Squire, K.: Design-based research: putting a stake in the ground. J. Learn. Sci. **13**(1), 1–14 (2004)
12. Linder, S.M., Ramey, M.D., Zambak, S.: Predictors of school readiness in literacy and mathematics: a selective review of the literature. Early Childhood Res. Pract. **15**(1) (2013). https://eric.ed.gov/?id=EJ1016152
13. Pham, H.: PyAudio: cross-platform audio I/O with PortAudio. Accessed 01 Aug 2019. https://people.csail.mit.edu/hubert/pyaudio/
14. Drew, J.: mpg321. Accessed 01 Aug 2019. https://mpg321.sourceforge.net/
15. MediaDevices: getUserMedia() method - Web APIs|MDN. https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getUserMedia
16. Thompson, M., Lin, G.C., Schoenfeld, I., Uz-Bilgin, C., Leech, K.: Taking advice from a virtual agent: usability of an artificially intelligent smart speaker app for parent and child storybook reading. In: Filipiak, D., Kalir, J.H. (eds.) Proceedings of the 2022 Connected Learning Summit, pp. 100–108. Carnegie Mellon University, ETC Press, Virtual (2022). https://doi.org/10.57862/tg5r-ck86

17. Lin, G.C., Schoenfeld, I., Thompson, M., Xia, Y., Uz-Bilgin, C., Leech, K.: "What color are the fish's scales?" Exploring parents' and children's natural interactions with a child-friendly virtual agent during storybook reading. In: Interaction Design and Children, pp. 185–195. ACM, Braga (2022). https://doi.org/10.1145/3501712.3529734
18. Smith, C., et al.: Interaction strategies for an affective conversational agent. In: Allbeck, J., Badler, N., Bickmore, T., Pelachaud, C., Safonova, A. (eds.) IVA 2010. LNCS (LNAI), vol. 6356, pp. 301–314. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15892-6_31
19. Smith, C., et al.: Interaction strategies for an affective conversational agent. Presence **20**(5), 395–411 (2011). https://doi.org/10.1162/PRES_a_00063
20. Boutte, G.S., Hopkins, R., Waklatsi, T.: Perspectives, voices, and worldviews in frequently read children's books. Early Educ. Dev. **19**(6), 941–962 (2008). https://doi.org/10.1080/10409280802206643