ScratchJr Demo: A coding language for Kindergarten

Amanda Strawhacker
DevTech Research Group
105 College Avenue
Medford, MA 02144
1-617-627-4490
amanda.strawhacker@tufts.edu

Melissa Lee
DevTech Research Group
105 College Avenue
Medford, MA 02144
1-617-627-4490
melissasc.lee@tufts.edu

Claire Caine
DevTech Research Group
105 College Avenue
Medford, MA 02144
1-617-627-4490
claire.caine@tufts.edu

Marina Bers
DevTech Research Group
105 College Avenue
Medford, MA 02144
1-617-627-4490
marina.bers@tufts.edu

ABSTRACT
This paper describes the ScratchJr research project, a collaboration between Tufts University's Developmental Technologies Research Group, MIT's Lifelong Kindergarten Group, and the Playful Invention Company. Over the past five years, dozens of ScratchJr prototypes have been designed and studied with over 300 K-2nd grade students, teachers and parents. ScratchJr allows children ages 5 to 7 years to explore concepts of computer programming and digital content creation in a safe and fun environment. This paper describes the progression of major prototypes leading to the current public version, as well as the educational resources developed for use with ScratchJr. Future directions and educational implications are also discussed.

Categories and Subject Descriptors
D.2.2 **[Software Engineering]**: Design tools and techniques – *evolutionary prototyping, user-centered design*, D.2.6 [**Software Engineering**]: Programming Environments – *graphical environments,* K.3.2 [**Computer and Information Science Education**]: Computer science education.

General Terms
Design, Documentation, Human Factors, Languages.

Keywords
Graphical Programming, Early Childhood, Education, STEM, Tablets

1. INTRODUCTION
For the past four years, the Developmental Technologies (DevTech) Research Group and the MIT Media Lab have been working with the Playful Invention Company to design a version of the Scratch programming environment (www.scratch.mit.edu) designed especially for young children, ages 5-7 years. In 2013, this team presented an initial prototype and a discussion of the design process involved in creating a computer programming experience for young children ScratchJr. The culmination of years of research, iterative development, and user testing in classrooms is the final version of ScratchJr, freely available on a variety of tablet platforms, along with a rich library of online teaching and learning resources to be used in classrooms and homes around the world.

Coding languages for children have gained considerable popularity in recent years, with federal education programs and private initiatives making computer science and technological literacy a priority for young children (www.code.org) [3, 10]. Since 2003, Scratch (www.scratch.mit.edu) has maintained a thriving online community of creators (mainly ages 8-16 years) who use the Scratch programming language to build and share original games, stories, videos, and other coded explorations [9]. Research with computer programming interventions in early childhood settings has shown that children as young as 5 years old can master fundamental programming concepts of sequencing, logical ordering, and cause-and-effect relationships [1, 5, 7, 8]. ScratchJr is a developmentally appropriate learning tool to introduce fundamental concepts of computer programming to children ages 5 to 7 years.

In addition to computer programming skills, ScratchJr is designed to foster the following target learning outcomes for young children: 1) domain-specific knowledge in areas of literacy and mathematics, 2) foundational cross-domain knowledge structures, such as prediction and classification, and 3) complex problem solving skills, including aspects of the engineering design process like planning and testing solutions. By combining a developmentally appropriate interface with a powerful, open-ended programming platform, ScratchJr allows children to achieve these learning goals at their own pace, as part of a self-directed process of design and creation.

2. EVOLUTION OF SCRATCHJR
Because the effectiveness of ScratchJr depends upon the ways that children understand and use it, user testing with teachers, parents, and children was pivotal to its development. Several versions and iterations were tested and adapted before the current version was

IDC '15, June 21-25, 2015, Medford, MA, USA
ACM 978-1-4503-3590-4/15/06.
http://dx.doi.org/10.1145/2771839.2771867

released to the public. Since its launch in late July, over 600,000 downloads have occurred worldwide.

2.1 Pilots and Prototypes

Early prototypes of ScratchJr were based on findings from using Scratch (v1.4) with early childhood classrooms and noting reactions and feedback [6]. From this baseline, new versions were developed that focused on principles of developmentally appropriate design, as well as critical learning outcomes for early elementary students. Three major prototypes emerged from the baseline explorations (see table 1).

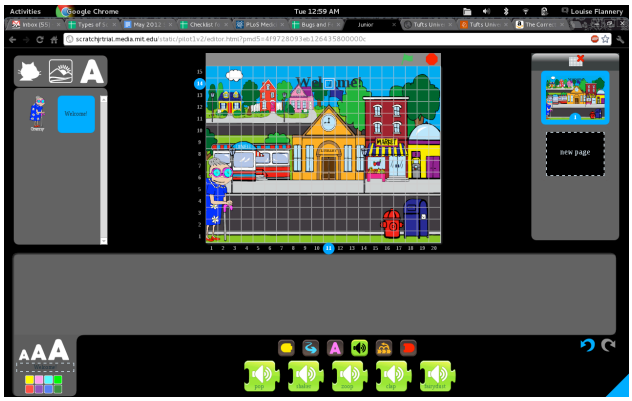

Figure 1. Alpha version of ScratchJr

Alpha, the initial prototype, was a web-based version that required children and teachers to log in to a private server to access saved work (see fig. 1). This version addressed overarching goals of scaling the Scratch programming environment to a younger audience (i.e. less text on screen, more inviting and colorful graphics, large programming blocks with simple commands, etc.) but much work was needed to streamline ScratchJr for use in classrooms. The username system was poorly received, since teachers had difficulty keeping track of students' login credentials and it was too easy for children to create new accounts. Most children ended up with multiple accounts and lost work. Mainly, teachers were frustrated that they were unable to view student work without logging into and out of dozens of student-made accounts, often multiple for each student. However, the most common request from teachers was a tablet-based version of

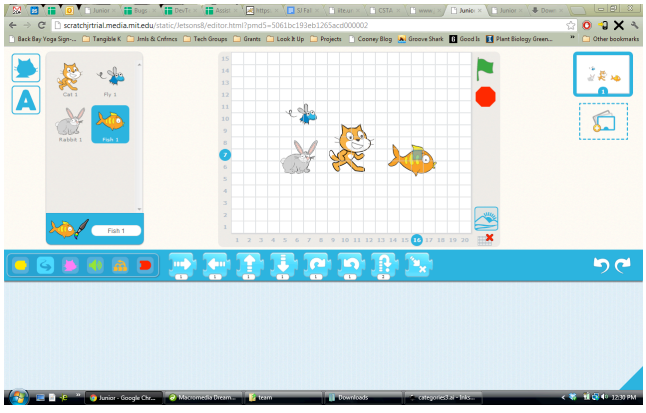

Figure 2. Beta version of ScratchJr

ScratchJr that did not require internet access. It is conceivable that this mandate was partially spurred by the growing popularity in 2011 and 2012 of Apple iPad tablets and touch-screen devices, but the beta research also showed that students had considerable difficulty manipulating mice and touchpads when using ScratchJr on computers. Additionally, many schools had slow internet connections that caused lags and errors as students worked, creating frustrations for both students and teachers.

The Beta version of ScratchJr was developed with classrooms in mind, and major changes focused on addressing challenges from the Alpha version (see fig. 2). To address the issue of teachers wanting to view student work, an experimental Admin Panel was created. This was a master-user webpage for teachers that grouped all student accounts and completed work in one location. While there were some successes with this method (for example, it was easy to for teachers to view all student work at once), the Admin Panel was overwhelming to use, as each classroom might have hundreds of projects by the end of a unit. The Beta was also released as a tablet interface for iPads, which did show improvements in children's ability to quickly and fluently create projects in ScratchJr. However, this version was actually a website optimized for mobile use, and not a native application. Classrooms still needed access to wireless internet in order to use ScratchJr, which ultimately became the biggest drawback of the Beta version.

2.2 User Testing with Children, Teachers, and Parents

Throughout the development of the major prototypes (and dozens of smaller updates and iterations in between), user testing was implemented as a core component of the ScratchJr research team's design process. Informal afterschool sessions, educator workshops, experimental classroom interventions, and at-home play sessions all provided valuable insight from parents, teachers, and young children about their experiences using ScratchJr These findings directly influenced the current design of the app.

Most ScratchJr user testing took place in classrooms and schools. This was a convenient way to try the software out with large numbers of children, as well as a useful exercise for capturing educator and administrator opinions about effective classroom learning tools. User testing focused heavily on educator focus groups and workshops, especially during the first year of the project. Twenty-six teachers over four years contributed to the design changes and classroom testing of ScratchJr. Many of their suggestions have survived to the current version, such as the idea to have one complete rotation of the "rotate character" block set at 12 steps because it corresponded with the numbers on an analog clock, a topic typically covered in first and second grade. Other requests, such as an "off switch" to hide portions of the block palette, required the research team to think carefully about the pedagogical philosophy behind the layout of ScratchJr. Similarly, parent feedback inspired many decisions such as the lack of internet links anywhere in the app (so children cannot accidentally get online) and the email-sharing feature, to send projects to family members or other devices.

Of course, user testing with children carried the most weight in design decisions, and K-2nd grade children constituted the largest user-testing group. Over 300 children participated in formal classroom interventions, semi-structured camps, and open-ended play sessions. Their feedback directly led to the creation or refinement of many features, including the camera functionality inthe paint editor, the sound recording feature in the Sound Blocks palette, the colorful highlighting of blocks as they are running within a program, and the drag-to-copy feature that allows characters and the code to be shared across multiple pages in a project. Throughout all of our research, the same data has been

Table 1. Evolution of Major Features in ScratchJr Prototypes

Features	Scratch v1.4 (Baseline)	Alpha (2012)	Beta (2013)	Current Version (2015)
Platform for use	Online, computers only	Online, computers only	Online, tablet and computers	Multi-platform tablet application, no internet necessary. Online version scheduled for release before 2016.
Sharing Projects	Online community, all content is freely available	Member center for each child. This became confusing for children (lost logins, etc.)	Admin center for teachers. This was overwhelming for educators, too many projects to log.	Email and AirDrop® capability
Personalization of Projects	Ability to import images and import or record sounds from online and from saved files on computer	Paint editor for characters. Pre-populated library of sounds	Same as Alpha	Paint editor enhanced with camera capability. Sound recording using microphone functionality.
Deleting Characters	Trash can in lower corner of canvas	Trash can in corner of screen. Children often confused about what could and could not be deleted	Trash can in corner character list. Design issue around deleting other objects (pages, on-canvas text, etc.)	Gesture-based delete function (press-and-hold). Applied to all editable objects, both on and off canvas.
Saving Projects	Saved as file to computer, no auto-save	Auto-save when exiting project	Same as Alpha	Auto-save when exiting project and when sharing project.
Guides and Tutorials	No in-software tutorials. Online forums, videos, and guides	No in-software tutorials. Printed activity guides	Same as Alpha	In-software tutorials, videos, and sample projects. Online curricula, activity guides, and instructional videos.

collected over and over in different settings: children and adults want ScratchJr to be as open-ended as possible, so that they can let their creations and ideas take center stage when learning to code.

2.3 Final Design: ScratchJr Release

The feedback and research from the Alpha and Beta phases of ScratchJr contributed to the creation of the Release Candidate version in 2014. Although there have been minor updates and bug-fixes since then, this version is very close to what is currently available for public download, and so for the duration of the paper, this version will be referred to as the "current version" (see table 1). The current version includes a number of revisions and updates, such as the use of the press-and-hold gesture to replace confusing delete icons and the development of a native tablet ScratchJr that does not require internet access. However, the core changes in the app focus on usability. For example, the username and admin panel systems were removed entirely in place of a simple one-device-to-one-device sharing model. Projects can now be shared via email or Apple's AirDrop feature. This was a good compromise for school use, where teachers only needed to view a few projects from each student, and for home use, where parents and guardians reported that they mainly wanted to be able to send one or two finished projects at a time. Additionally, this removed the issue of children forgetting usernames.

Another major goal for the current version is broad availability. It is imperative to the research team that ScratchJr reach as many children as possible, regardless of location, access to technology, etc. To that end, a core focus since the release of the current version has been to port to different platforms, especially Android tablets and most web-browsers.

3. WHAT'S NEXT FOR SCRATCHJR?

Since the initial public release of ScratchJr, there have been over 600,000 downloads from the Apple App store alone, and the response from ScratchJr users has been overwhelmingly positive. Now, the ScratchJr research team is focused on researching ways that ScratchJr and similar technologies can be effectively introduced in a variety of learning settings. This goal involves investigating what educators need to successfully introduce computer programming content, as well as creating and disseminating useful teaching and learning resources, such as curricula, teaching materials, and assessments. Additionally, new research questions will be investigated such as how children think and communicate about their own coded creations, and how collaborative programming among family members can influence experiences with programming in ScratchJr.

3.1 Teaching Approaches and Resources

To make the teaching experience as easy as possible, the research team has published extended curricula, short activities, and even experimental programming assessment materials for free download on the ScratchJr website (www.scratchjr.org). Our goal is to invite educators and parents to compare perspectives and approaches for introducing computer programming to young children.

One major area of growth in the coming year will be the resource library. As previously mentioned, the inclusion of in-app tutorials and interface guides was a huge step in finalizing the

current version of ScratchJr. Research with teachers and parents has shown that teacher's pedagogical beliefs are an important factor for the success of introducing any technological tool to young learners [2]. There is also considerable evidence that teaching styles and adult attitudes influence the learning culture of a classroom or home environment [4]. In future studies, we hope to understand how different teaching methods and approaches can influence student learning. This will help us determine the effectiveness and value of various teaching resources when introducing ScratchJr to new students.

4. CONCLUSION

This paper presents the evolution of a technology designed to introduce computer programming to young children. The design process leading to the final public release is outlined in an effort to contribute to the growing body of research about designing tablet applications for early childhood. There are unique challenges in researching learning outcomes with a technology while simultaneously developing the technology. Through beta testing with children, educators, and parents, and continued research around children's programming approaches, the ScratchJr programming language has been iteratively honed and redesigned. It is the hope of the researchers that now, ScratchJr can provide a powerful, open-ended platform for children to create meaningful coding explorations, and a rich area for the research community to explore and understand children's learning processes when learning to code.

5. ACKNOWLEDGMENTS

This project is supported by National Science Foundation grant DRL-1118664 and the Code-to-Learn Foundation. The researchers also thank the many teachers, parents, and children who contributed to the user testing described in this paper.

6. REFERENCES

[1] Bers, M. U. (2008). *Blocks to Robots: Learning with technology in the early childhood classroom*. Teachers College Press, NY, NY.

[2] Chen, C. (2008). Why do teachers not practice what they believe regarding technology integration? The Journal of Educational Research, 102(1), 65-75. doi:http://dx.doi.org/10.3200/JOER.102.1.65-75

[3] Department for Education. (2013). The National Curriculum in England: Framework document. London: The Stationery Office.

[4] Ertmer, P. A., Ottenbreit-Leftwich, A., & York, C. S. (2007). Exemplary technology-using teachers: Perceptions of factors influencing success. Journal of Computing in Teacher Education, 23(2), 55-61. Retrieved from http://search.proquest.com.ezproxy.library.tufts.edu/docview/742872292?accountid=14434

[5] Fessakis, G., Gouli, E., and Mavroudi, E. (2013). Problem solving by 5-6 year old kindergarten children in a computer programming environment: A case study. *Computers & Education, 63*, pp. 87-97. DIO=10.1016/j.compedu.2012.11.016

[6] Flannery, L.P., Kazakoff, E.R., Bontá, P., Silverman, B., Bers, M.U., and 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. DOI=10.1145/2485760.2485785

[7] Kazakoff, E.R., & Bers, M.U. (2011). The Impact of Computer Programming on Sequencing Ability in Early Childhood. *Paper presented at American Educational Research Association Conference (AERA)*, 8 - 12 April, 2011, Louisiana: New Orleans.

[8] Kazakoff, 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. DOI=10.1007/s10643-012-0554-5.

[9] Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), pp. 60-67. DOI=10.1145/1592761.1592779

[10] US Department of Education, Office of Educational Technology (2010). Transforming American education: Learning powered by technology. Draft National Educational Technology Plan 2010. Washington, DC: Author.