BEST OF BOTH WORLDS: ISSUES OF STRUCTURE AND AGENCY IN COMPUTATIONAL CREATION, IN AND OUT OF SCHOOL

Karen A. Brennan

B.Sc. Computer Science and Mathematics (2003)
B.Ed. Computer Science and Mathematics (2005)
M.A. Curriculum Studies (2007)
The University of British Columbia

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Media Arts and Sciences at the Massachusetts Institute of Technology.

February 2013

AUTHOR

Karen A. Brennan
Program in Media Arts and Sciences
October 29, 2012

CERTIFIED BY

Mitchel Resnick
LEGO Papert Professor of Learning Research
Program in Media Arts and Sciences
Thesis Supervisor

ACCEPTED BY

Patricia Maes
Associate Academic Head
Program in Media Arts and Sciences

BEST OF BOTH WORLDS: ISSUES OF STRUCTURE AND AGENCY IN COMPUTATIONAL CREATION, IN AND OUT OF SCHOOL

Karen A. Brennan

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning, on October 29, 2012, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Media Arts and Sciences.

ABSTRACT

We live in a computational culture – a culture in which we are surrounded by computational systems and interfaces, from social networks to banking infrastructure, to entertainment platforms, to transportation systems. This culture introduces new expectations and new opportunities for learning, creating new demands for *what* to learn and offering new possibilities for *how* to learn.

In this dissertation, I adopt a predominantly qualitative approach to exploring learning in computational culture, studying how the Scratch programming environment and online community are employed to support learning both in and out of school. To this end, I conducted interviews with 30 kids working with Scratch at home and 30 teachers working with Scratch in K-12 classrooms to develop descriptions of computational creation in these two settings.

Using a theoretical framework of agency and structure, I analyze how the at-home and school-classroom contexts enable – or constrain – young people's agency in computational creation. Despite common assumptions that at-home learning is necessarily low-structure/high-agency and that at-school learning is necessarily high-structure/low-agency, I argue that structure and agency need not be in opposition. Designers of learning environments should explore intermediate possibilities, finding ways to employ structure in the service of learner agency.

THESIS SUPERVISOR

Mitchel Resnick
LEGO Papert Professor of Learning Research
Program in Media Arts and Sciences

BEST OF BOTH WORLDS: ISSUES OF STRUCTURE AND AGENCY IN COMPUTATIONAL CREATION, IN AND OUT OF SCHOOL

Karen A. Brennan

THESIS READER

Barry J. Fishman
Associate Professor of Educational Studies and Learning Technologies
The University of Michigan
School of Education and School of Information

BEST OF BOTH WORLDS: ISSUES OF STRUCTURE AND AGENCY IN COMPUTATIONAL CREATION, IN AND OUT OF SCHOOL

Karen A. Brennan

THESIS READER

Ethan Zuckerman
Principal Research Scientist
Director of the MIT Center for Civic Media
MIT Media Lab

This material is based upon work supported by the National Science Foundation under Grant No. 1019396. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Acknowledgements

On my last day of kindergarten, I refused to get on the bus that was to take me home. The thought of summer break, away from the exciting activities in this magical place, made me indescribably sad. I was inconsolable, crying and clinging to the leg of my kindergarten teacher, Mrs. Marshall, until my mother (who had been called by the school principal) arrived to take me home.

Now, almost 30 years later, I approach another last day of kindergarten, having nearly completed my time as a graduate student in the Lifelong Kindergarten research group at the MIT Media Lab. I am again filled with the sadness of leaving a magical place. But this sadness is easily surpassed by the gratitude and appreciation that I feel for the many people who have contributed to the work described here and to the wonder that I have experienced in the past five years.

I would like to begin by thanking my advisor, Mitch Resnick. Mitch – I would not be here without you. I cannot sufficiently express my gratitude for having had the opportunity to be your student. I can only hope that I am able to support others in the ways you have supported me.

I would also like to extend my most sincere appreciation to my readers, Barry Fishman and Ethan Zuckerman. Both Barry and Ethan are an amazing combination of always-encouraging and always-provoking – a combination that is disarmingly productive. Barry – thank you for your incredible generosity over the past five years. You are so giving with your time, advice, opportunities, and beignets. And thank you for being the first to ask me the question that has shaped my work in every way: "Why Scratch in schools?" Ethan – thank you for always being such a careful and thoughtful reader, highlighting areas to pursue in greater depth and encouraging criticality (an image of you reading with Wolverine claws is now deeply embedded in my imagination).

The magic of the Media Lab has been central to my work – I couldn't have done this work, in this way, anywhere else. I've had the opportunity to meet some of my closest collaborators and friends:

Stephanie Gayle, Shani Daily, and Sajid Sadi. I've had the opportunity to work with inspiring faculty members, including Sherry Turkle and Judith Donath. I've had the opportunity to belong to an amazing research group, learning and playing alongside my Lifelong Kindergarten colleagues, and a wonderful community of Media Lab students, staff, and faculty. A special thanks to Linda Peterson and to Kevin Davis, Cornelle King, Ryan Walsh, Bill Lombardi, and Greg Tucker – thank you for your constant support and help in navigating all things Media Lab. I will miss seeing you each day.

Numerous interns have contributed to my research over the past five years. Thanks to my Harvard graduate student interns – Jeanne Wellings, Robyn Bykofsky, Michelle Chung, Ashley Lee, Amanda Valverde, Joe Prempeh, Linda Qian, Anushka Paul, Krista Shapton, Mydhili Bayyapunedi, Gracie Elqura, Mylo Lam, Aaron Morris, Vanessa Gennarelli, Jen Lavalle, Alex Schoenfeld, and Vanity Gee – as well as my MIT and Yale undergraduate interns – Mónica Oliver, Laura Wacker, Crystal Noel, and Jane Long. You have contributed to every aspect of this research. It would not have happened without your hard work, persistence, passion, and endless enthusiasm. A special thank-you to Michelle, who started as an intern in 2009 and joined Team ScratchEd full-time as research staff in 2010. It has been a pleasure developing the ScratchEd research program with you. I don't express it often enough, but I deeply appreciate the time, energy, and kindness that you dedicate to the work.

There is life beyond the Lab, however, and I'd like to thank my family for reminding me of that. In particular, my sister, Shannon Brennan, and my brother-in-law, Dan Montsion, have been constantly supportive, providing much-needed fun and offering their understanding when life was too hectic.

Finally, I would like to thank my partner, Jeff Hawson. Jeff – I really wanted this big fish and you helped me catch it. I love you so much for never questioning why this fish might be important, and for reminding me why I was so excited about the fish when things were hard and I doubted that it could be caught. We did it!

To all of the young Scratchers and Scratch educators, who were so generous in sharing their time and their ideas.

Contents

Chapter 1

COMPUTATIONAL CULTURE

In this chapter, I describe the new demands on and new opportunities for learners in a computation-centric culture. I introduce my research questions, which focus on the role of structure in supporting the agency of computational creators, in both out-of-school and in-school settings.

We live in a computational culture – a culture in which we are surrounded by computational systems and interfaces, from social networks to banking infrastructure, to entertainment platforms, to transportation systems. This culture introduces new expectations and new opportunities for learning, creating new demands for *what* to learn and offering new possibilities for *how* to learn. Two recent events illustrate these expectations and opportunities.

WHAT TO LEARN

On June 21, 2012, the microblogging platform Twitter was unavailable for several hours. Twitter's Vice-President of Engineering, Mazen Rawashdeh, posted an event report and apology on the company's blog, with the title "Today's turbulence explained".[1]

> Not how we wanted today to go. At approximately 9:00am PDT, we discovered that Twitter was inaccessible for all web users, and mobile clients were not showing new Tweets. We immediately began to investigate the issue and found that there was a cascading bug in one of our infrastructure components. This wasn't due to a hack or our new office or Euro 2012 or GIF avatars, as some have speculated today. A "cascading bug" is a bug with an effect that isn't confined to a particular software element, but rather its effect "cascades" into other elements as well. One of the characteristics of such a bug is that it can have a significant impact on all users, worldwide, which was the case today. As soon as we discovered it, we took corrective actions, which included rolling back to a previous stable version of Twitter. We began recovery at around 10:10am PDT, dropped again around 10:40am PDT, and then began full recovery at 11:08am PDT. We are currently conducting a comprehensive review to ensure that we can avoid this chain of events in the future. For the past six months, we've enjoyed our highest marks for site reliability and stability ever: at least 99.96% and often 99.99%. In simpler terms, this means that in an average 24-hour period,

1 http://blog.twitter.com/2012/06/todays-turbulence-explained.html

> twitter.com has been stable and available to everyone for roughly 23 hours, 59 minutes and 40-ish seconds. Not today though. We know how critical Twitter has become for you – for many of us. Every day, we bring people closer to their heroes, causes, political movements, and much more. One user, Arghya Roychowdhury, put it this way:
>
> *"OMG..twitter was down....closest thing to living without oxygen for most of us...."*
>
> It's imperative that we remain available around the world, and today we stumbled. For that we offer our most sincere apologies and hope you'll be able to breathe easier now.

In the hours and days following the service interruption, the media coverage of the event demonstrated both the dependence that many people have on Twitter – as evidenced by articles titled *Twitter users frantic as site crashes for hours* (Telegraph), *A rough day for Twitter addicts* (CNET), *Outage hurts Twitter more than it hurts you* (Wired News), *Twitter crashes hard, Internet freaks out* (CNNMoney) – and a lack of understanding about how the service functions, in general, and what caused this particular period of disruption – as evidenced by articles such as *Twitter suffers sustained outage in hacker attack* (Reuters), *Twitter says outage wasn't hackers or Euro 2012, but a software fault* (The Guardian), *Twitter says bug caused site to crash but was it really a hacker attack?* (Daily Mail), *Twitter blames 'bug' for outage* (Wall Street Journal), *Twitter had to use older version of site to function Thursday* (Los Angeles Times).

I learned about the Twitter disruption the day after it occurred, while watching four morning-show hosts on CNN. Their exchange illustrated that not everyone was satisfied with Twitter's "explanation".[2]

> We know there are some serious things facing this nation. We have high unemployment numbers, people losing homes. There is one bright spot in the world, though – Twitter's back, thank god.

2 http://www.cnn.com/video/#/video/bestoftv/2012/06/22/exp-point-get-real-twitter-crash.cnn

> 11:59am, just before noon yesterday, the microblogging site crashed. It was so bad that they never even put up the little "failwhale" that you get. Twitter returned around 1pm, less than an hour later, though, crashed again. Around 3 in the afternoon, Twitter's PR account tweeted that the issue was caused by a "cascading bug" – whatever that means.
>
> (Co-host) *I know – that's my new excuse for anything now.*
>
> Oh, [*with mock exasperation*] it's a cascading bug…

The host's "whatever that means" response to Twitter's "cascading bug" explanation was striking. The tone was ostensibly playful, a metaphoric eye-roll in response to yet more of the computational jargon that increasingly seeps into everyday conversation. But just underneath the playfulness, it wasn't hard to sense some frustration, dismissiveness, and resignation – the expressions of a helplessness often felt in our technologically saturated world, that if *only* we had greater command of and fluency with computation, all of the promises made about technology improving our personal, professional, and public lives could be fulfilled.

Most people, confronted daily by computational culture, experience some degree of this helplessness. This reality has given rise to a sense of urgency – expressed by a variety of sources from computer science education researchers (e.g., Guzdial & Forte, 2005) to literary theorists (e.g., Hayles, 2005) to government agencies (e.g., Chopra, 2012) – that everyone should be able to fully participate in computational culture. The urgency stems, in part, from a concern that unless we understand how to actively participate in computational culture, we risk being controlled by it.

> Everyday life is increasingly regulated by complex technologies that most people neither understand nor believe they can do much to influence. The very technologies they create to control their life environment paradoxically can become a constraining force that, in turn, controls how they think and behave. (Bandura, 2001, p. 17)

As Appadurai (1996) observed (about the world of advertising, though it applies equally well to the culture of computation), we will find that we are no longer "actors", if we ever were, and that we have become "choosers" instead (p. 42).

There is significant debate about what constitutes an acceptable standard of participation in computational culture. For example, does participation require only that one is able to use computational applications? Further, the name given to this participation and knowing varies, having taken on several different forms over the past decade – "21st century skills", "IT fluency", "technological literacy", "digital literacy", and more recently "computational thinking" (Hargittai, 2009; ITEA, 2007; National Research Council, 1999; National Research Council, 2010; The Partnership for 21st Century Skills, 2009; Wing, 2006).

The position I adopt is that to fully participate in computational culture, one needs to develop as a *computational creator*, and that *learning how to program* is a particularly rich activity for supporting this development and ensuing participation. Computational creators are familiar with certain computational concepts, such as sequences, loops, and variables. Computational creators are also familiar with computational practices, such as being able to remix and reuse others' code, or being able to abstract and modularize ideas. Furthermore, and perhaps most importantly, computational creators develop new perspectives on computation, "certain ways of understanding the world, … kinds of knowing that derive from a computer culture" (Papert, 1993, p. 51). With these new perspectives, an individual can see how computation can be used to express ideas and solve problems, to enable new connections between people, and to empower people to ask questions about computational culture.

Knowing how to program empowers people to participate as creators – not just consumers – in computational culture. We take computational culture for granted at the Media Lab, a place where people are able to participate as sophisticated readers and writers of computation. But most people do not participate as writers in computational culture (or even understand what that might imply) and often even struggle as readers. This gap between reading and writing is often compounded for young people, with narratives of young people as "digital natives" leading to expectations about assumed-to-exist abilities that are, in fact, unfulfilled (Selwyn, 2009). Understanding programming and understanding code is a critical layer of modern culture – and should not "remain the exclusive concern of computer programmers and engineers" (Hayles, 2005, p. 61). As Rushkoff (2010) argued, if we leave programming to a select group, "we risk being programmed ourselves" (p. 133).

HOW TO LEARN

On May 2, 2012, Director of MIT's Computer Science and Artificial Intelligence Laboratory Anant Agarwal, MIT President Susan Hockfield, Harvard President Drew Gilpin Faust, MIT Provost Rafael Reif, and Harvard Provost Alan Garber met with members of the media to announce edX, a new Harvard-MIT venture in online education. Agarwal welcomed the group and began his remarks by talking about a "revolution" in education, made possible by computational culture.[3]

> There is a revolution brewing in Boston and beyond. It does not have to do with tea. It does not have to do with the Boston Harbor. It does not have to do with guns and it does not have to do with the sword. Instead, this revolution has to do with the pen and the mouse. Online education, it is revolutionary. Online education will change the world. In a prototype course that we're offering as we speak, the number of students around the world that are taking it is insane, 120,000 students around the world. Online education is disruptive. It will completely change the world. Students from Tunisia, Pakistan, India, New Zealand, Australia, Colombia, the USA, Canada, all working on learning, all collaborating and working together. Students creating small groups in Cairo, meeting in tea shops and discussing, guess what, technology and education and learning, humanities, sciences.

3 http://www.youtube.com/watch?v=7pYwGpKMXuA

> It is unbelievable. Online education truly has the potential to change the world. Modern technology such as the Internet, cloud computing, computing, machine learning, and so on are really coming together to make it possible for us to offer online education at a massive scale around the world.

The idea of learning online was not what I found to be the most interesting or novel aspect of the edX announcement, yet another in a seeming multitude of similar announcements and initiatives. Rather, it was that two of the world's most highly-esteemed, traditional educational institutions were significantly investing in these online activities and in the new opportunities for learning made possible by computational culture – activities and opportunities that have the potential to fundamentally disrupt the centrality of the "school" and traditional educational settings.

For the past hundred years, school has been assumed to be the primary site of learning. But increasingly, as evidenced by the edX announcement, school is no longer the *only* place of learning, a trend particularly supported by the use of computer networks (Collins & Halverson, 2009; Ito et al., 2009; Thomas & Brown, 2011). Learners have new control and new responsibilities in these new places of learning.

> We grew up with the idea that learning means taking courses in school. … [T]he identification of education with schooling is slowly unraveling, as new technologies move learning outside of school's walls. In some sense, the divorce of schooling and learning may take us back to an era where individuals negotiate their own learning experiences. (Collins & Halverson, 2009, p. 129)

Computational culture moves beyond needing to "funnel all educational programs through the teacher" to "provid[ing] the learner with new links to the world" (Illich, 1971, p. 73). Learning – whether it happens in school, at work, at home, with friends, family, colleagues, or strangers – should situate the learner at the center of the process, participating in a way that respects and supports their *agency* as learners.

Agency is multiply defined, and has been associated with a wide variety of ideas, including "self-hood, motivation, will, purposiveness, intentionality, choice, initiative, freedom, and creativity" (Emirbayer & Mische, 1998, p. 962). Here, following Bandura (2001) and Martin (2004), my working definition of learner agency is *a learner's ability to define and pursue learning goals*, which enables him or her "to play a part in their self-development, adaptation, and self-renewal with changing times" (Bandura, 2001, p. 2).

agency
A learner's ability to define and pursue learning goals.

Being actively involved in defining and pursuing one's learning goals, rather than as passive consumer of externally-imposed educational objectives, is simultaneously an *independent* role and a *connected* role for learners. It is an independent role in that learners have the freedom and responsibility to identify what they care about, what they are interested in, and what they will need to achieve their goals. It is a connected role in that learners, enabled by network technologies in computational culture, have the opportunity and expectation to form links to resources and with other learners, to seek out particular structures for support and scaffolding of their learning.

But although we are entering an era in which learner agency is of particular importance, there is uncertainty about the processes involved in fostering young people's agency as learners. In particular, a central question about supporting learner agency is the role of *structure*.

Like agency, structure is a word that, despite its pervasiveness, eludes crisp definition, appearing "in social scientific discourse as a powerful metonymic device, identifying some part of a complex social reality as explaining the whole" (Sewell, 1992, p. 2). Following Sewell's (1992) theorization (based on the work of Giddens and Bourdieu), structure is manifested through *rules, roles, and resources, both explicit and assumed*. In a classroom, for example, a lesson plan is a resource that serves as explicit structure and the teacher as expert is an often-enacted assumed role.

structure
Rules, roles, and resources, both explicit and assumed.

The tension between agency and structure – and the implications for the design of learning environments – has been an ongoing discussion in educational research (e.g., Craig, 1956; Anthony, 1973; Perkins, 1991). One extreme reaction to the history of learning environments with low learner agency has been to completely remove any structure, as it might unduly impinge on the agency of the learner (Kafai, 2006). In the context of digital media and learning, this reaction against structure is fueled by assumptions about digital natives (presumed to already know everything or to be able to learn it independently) and by assumptions about schools (seemingly unchanged in the past 100 years and ill-equipped to support the agency-centered learning necessary in the 21st century).

But *eliminating* structure does not ideally support learning, as structure can benefit learners in their development as individuals capable of defining and pursuing learning goals. Scardamalia and Bereiter (1991), while supporting environments of high learner agency, highlighted the value provided by the structure of other learners and of tools for identifying what is known and not known, and cautioned against "romanticizing the idea of the child as independent knowledge builder" (p. 40). Kirschner, Sweller, and Clark (2006) argued that a lack of structure disadvantages novice learners, who benefit from "direct, strong instructional guidance" in defining and achieving their learning goals (p. 83). Mayer (2004), in examining the particular context of how young people develop as computational creators, argued that a review of the youth computer programming literature of the mid-1980s (mostly connected to the low-structure aspirations of the Logo programming movement) illustrated "the failure of pure discovery as an effective instructional method" (p. 17).

Further, aspirations for removing all structure are misplaced, as they confuse the fundamental relationship between agency and structure. A "structureless" environment is not actually an option, because it does not exist – agency and structure are not in opposition, they mutually constitute each other (Bandura, 2001; Buckingham & Sefton-Green, 2003; Emirbayer & Mische, 1998; Freeman, 1972; Giddens, 1984; Schwartz & Okita, 2009). The connections between

Agency and structure are not in opposition. They mutually constitute each other.

agency and structure are elaborated in Giddens' (1984) structuration theory. We have agency through structure, and we have structure through agency –

> Theorizing about human agency and collectivities is replete with contentious dualisms that social cognitive theory rejects. These dualities include personal agency versus social structure, self-centered agency versus communality, and individualism versus collectivism. The agency-sociostructural duality pits psychological theories and sociostructural theories as rival conceptions of human behavior or as representing different levels and temporal proximity of causation. Human functioning is rooted in social systems. Therefore, personal agency operates within a broad network of sociostructural influences. For the most part, social structures represent authorized systems of rules, social practices, and sanctions designed to regulate human affairs. These sociostructural functions are carried out by human beings occupying authorized roles. (Bandura, 2001, p. 14)

In applying structuration theory to thinking about how people learn and to the design of learning environments in computational culture, we see that structure need not be in opposition to agency. As structure "is always both constraining and enabling" (Giddens, 1984, p. 25), there exists the potential, with careful design, to employ structure in a way that amplifies learner agency. Further, different structures will constrain and enable in different ways, and different learning environments have different structures. Accordingly, designers of learning environments need to consider the affordances of the settings in which they work and how to design for learner agency in relationship to structure.

STUDYING LEARNING IN COMPUTATIONAL CULTURE

Developed by the Lifelong Kindergarten research group at the MIT Media Lab, Scratch – which is both an authoring environment for programming interactive media projects and an online platform for sharing those projects – has served as a particularly rich context for supporting both the *what* and *how* of learning in computational culture. Since Scratch's launch in May 2007, hundreds of thousands of young people have downloaded the Scratch authoring environment, developed millions of interactive media projects, and shared their creations with other young learners via the online community. Scratch has also been increasingly used in formal learning environments, such as K-12 classrooms, with teachers using Scratch with learners across a range of ages and across a variety of curricular areas.

But a tool cannot dictate how it is used in a particular environment, despite the intentions or aspirations of the tool's designer (Scardamalia & Bereiter, 1991). As such, the use of Scratch has been significantly influenced by the structures present in individual learning environments, which has led to different learner activities and varying levels of learner agency. As a member of the group that develops Scratch and as the lead for educational outreach efforts with Scratch, I have been fascinated by the ways in which Scratch is employed in different settings to support young people's development as computational creators. In particular, I have focused on two settings that would seem to represent extremes along an agency/structure spectrum – the use of Scratch by kids at home to support their own learning (a setting that is assumed to be high agency and low structure) and the use of Scratch by teachers in K-12 classrooms to support the learning of their students (a setting that is assumed to be low agency and high structure).

The tension between structure and learner agency (and how it varies across settings) is of central importance in the design of all learning environments – and has preoccupied me as a designer and researcher. This tension lacks both theoretical and empirical attention in the research literature, particularly in the context of digital media learning and computational creation. As Emirbayer and Mische (1998) observed, there is a need for work that is grounded in experience – "the empirical challenge becomes that of locating, comparing, and predicting the relationship between different kinds of agentic processes and particular structuring contexts of action"

(p. 1005) – a need also echoed more recently (Damşa, Kirschner, Andriessen, Erkens, & Sins, 2010; Martin, 2004; Pea, 2004).

Drawing on the contexts of young people working with Scratch at home and teachers working with Scratch in K-12 classrooms, my work is guided by two central research questions, as a way of developing understandings about the creative activities and agentic processes of young computational creators:

1. How do out-of-school and in-school learning environments support the activities of computational creators?

2. Within these learning environments, how does structure enable, rather than constrain, the agency of young computational creators?

My explorations of these questions are organized into seven chapters.

In Chapter 1, *Computational Culture,* I describe the new demands on and new opportunities for learners in a computation-centric culture. The chapter is organized into three parts. The first part – *What To Learn* – describes computational culture and how computational creation supports participation in the culture. The second part – *How To Learn* – describes the new opportunities for learning in computational culture. Here, I introduce *agency* (defined as *a learner's ability to define and pursue learning goals*) and *structure* (defined as *rules, roles, and resources, both explicit and assumed*), key theoretical ideas that guide this work. I argue for the centrality of agency in learning environments, and for considering agency and structure as mutually constitutive (rather than in opposition). The third part – *Studying Learning In Computational Culture* – describes the context for my research. I introduce my research questions, which focus on the role of structure in supporting the agency of computational creators, in both out-of-school and in-school settings.

In Chapter 2, *Contexts*, I introduce the two main settings for this thesis: kids working with Scratch at home and teachers working with Scratch in K-12 classrooms. The chapter is organized into three parts. The first part – *Scratch* – describes the origins and features of the Scratch authoring environment and the Scratch online community. The second part – *ScratchEd* – describes the origins and features of the ScratchEd online community and other educational outreach activities. The third part – *Constructionist Aspirations* – provides theoretical context for these settings. I describe my central assumptions about (and aspirations for) learning – how to support kids and teachers in designing, personalizing, sharing, and reflecting, ideas grounded in constructionist theories of learning. I end with a reflection on constructionist perspectives on agency and structure.

In Chapter 3, *Conversations*, I describe my process for developing understandings and descriptions of the out-of-school and in-school learning environments. For each setting, my primary source of data is conversations with the people primarily responsible for navigating and negotiating the various structures of the learning environment. For the out-of-school setting, I focus on conversations with kids working with Scratch at home, through the Scratch online community. For the in-school setting, I focus on conversations with teachers working with Scratch at school, in K-12 classrooms. The chapter is organized into two parts. The first part – *Collecting* – explains the data collection procedure, describes the backgrounds of the people I spoke with and their representativeness, and discusses the de/limitations of the data collection process for each of the two settings. The second part – *Analyzing* – explains my analytical approach, describing how the theoretical framework of agency and structure serves as a foundation for subsequent analysis.

In Chapter 4, *Kids*, I describe how kids talked about their goals and aspirations for creating and connecting with Scratch – and the tensions that emerged when striving to achieve those goals. The chapter is organized thematically into four sections, based on issues that kids highlighted as important and that recurred across the conversations. The first theme – *Enjoying Freedom* – highlights the freedom that kids enjoyed both in product (the diversity of creation) and in process (the responsibility of defining and pursuing their own challenges). The second theme – *Getting Stuck* – focuses on the problems kids encountered during their open-ended creative design activities. The third theme – *Making Progress* – outlines the various strategies, both individual and social, that kids employed to overcome their creative obstacles and to develop as computational creators. The fourth and final theme – *Finding Audience* – shares the importance kids attribute to (and the difficulties they face in) seeking attention and finding audience for their creative work.

In Chapter 5, *Teachers*, I describe tensions that teachers identified between their aspirations and the actualities of implementing Scratch in the classroom. The chapter is organized thematically into four sections. The first theme – *Supporting Problem-Solvers* – unpacks teachers' motivations for working with Scratch. The second theme – *Negotiating Open-Endedness* – explores the challenges teachers face when trying to implement open-ended design within the structure of the classroom. The third theme – *Building Culture* – outlines strategies that teachers have developed in response to the challenges that accompany open-ended design activities. The fourth and final theme – *Legitimizing Learning* – describes the challenges teachers experience when trying to understand and explain the learning that is taking place in their learning environments.

In Chapter 6, *Agency/Structure*, I focus on the relationship between agency and structure, as manifested in the learning environments described in Chapter 4 and Chapter 5. This chapter is organized into two parts. The first part – *In The Wild* – revisits the experiences of kids working with Scratch at home. I describe the structures encountered by kids in the Scratch online community, from the perspective of these young, primarily self-managing learners. The second part – *In Classrooms* – revisits the experiences of teachers working with Scratch at school. I describe the structures encountered by kids and teachers in K-12 classrooms, from the perspective of the teachers who are responsible for designing these learning environments. For each setting, I identify the sources of structure encountered, accessed, and adapted, and discuss how those structures enabled or constrained the agency and activities of kids and teachers.

In Chapter 7, *The Best of Both Worlds*, I connect the threads from the previous chapters, arguing that designers of agency-supporting learning environments, rather than setting structure in opposition to agency, should judiciously employ structure in order to amplify agency. This chapter is organized into two parts. The first part – *Either-Or* – illustrates, with a story, how too much and/or too little structure can inhibit learner agency. The second part – *Intermediate Possibilities* – offers strategies for designers of learning environments, suggesting opportune ways of introducing structure in the service of learner agency.

In the epilogue, *Life After Scratch*, I reflect on the future from three perspectives – as a researcher, as a designer, and as a learner.

Chapter 2

CONTEXTS

In this chapter, I introduce the two main settings for this thesis: kids working with Scratch at home and teachers working with Scratch in K-12 classrooms. I also describe how constructionist approaches to learning influence my perspective on agency and structure.

Developing as a computational creator through programming, thereby enabling one to participate as a writer and creator in computational culture, has not been possible for many people, due to a lack of accessible *tools*, personally meaningful *activities*, and supportive *others*. People need *tools* that make it easy to get started. But programming languages have historically been difficult to use, involving specialized syntax that is unforgiving of even the smallest error, such as a missing semicolon (Kelleher & Pausch, 2005). People need meaningful *activities* that make it worthwhile to participate. But computer science education has long been criticized for developing learning environments and activities that are disconnected from the passions and interests of young people, or only appealing to a very narrow subset of young people (Resnick & Silverman, 2005; Turkle & Papert, 1990). Finally, people need access to *others* to support their learning. But despite increased awareness of computational creation (from events like Maker Faire, movies like The Social Network, and initiatives like Codecademy's Code Year), programming generally has an unflattering public image – something that is done by socially awkward people, often men, who are uninterested in interacting and working with others (Klawe, Whitney, & Simard, 2009; WGBH & ACM, 2009). Along all three of these dimensions (tools, activities, others), computer science and programming have a bad reputation – so it is unsurprising that there is a cultural devaluing of these activities and declining participation (Carter, 2006; CSTA, 2006; Foster, 2005; Maillet & Porta, 2010).

In response to these needs, the Lifelong Kindergarten research group has been developing Scratch. Scratch was designed to provide an easy-to-use authoring tool, a context for personally meaningful activities, and a setting for interacting with others, enabling young people to engage in computational creation. But design aspirations do not always align with actual use – and the interestingness and complexity of how Scratch is used to support learners emerges from the different settings in which it is employed.

In this chapter, I introduce the two main contexts for this thesis: kids working with Scratch at home and teachers working with Scratch in K-12 classrooms. I describe these contexts from my perspective as a designer, both as a member of the Scratch Team (supporting the development of the Scratch authoring environment and online community) and as lead for the ScratchEd Team (supporting the development of the ScratchEd online community and other initiatives for teachers).

The chapter is organized into three parts. The first part – *Scratch* – describes the origins and features of the Scratch authoring environment and the Scratch online community. The second part – *ScratchEd* – describes the origins and features of the ScratchEd online community and other educational outreach activities. The third part – *Constructionist Aspirations* – provides theoretical context for these settings. I describe my central assumptions about (and aspirations for) learning – how to support kids and teachers in designing, personalizing, sharing, and reflecting, ideas grounded in constructionist theories of learning. I end with a reflection on constructionist perspectives on agency and structure.

SCRATCH

Logo[1], a programming language that was initially developed in the late 1960s by Seymour Papert and colleagues at MIT, in cooperation with the technology firm Bolt, Beranek and Newman, is a significant predecessor to and intellectual inspiration for Scratch. In particular, Papert's vision for the types of relationships to expect and encourage between young people and computers still resonates today.

> In most contemporary educational situations where children come into contact with computers the computer is used to put children through their paces, to provide exercises of an appropriate level of difficulty, to provide feedback, and to dispense information. The computer programming the child. In the LOGO environment the relationship is reversed: The child, even at preschool ages, is in control: The child programs the computer. (Papert, 1980, p. 19)

1 For a history of the development and proliferation of Logo, I recommend visiting http://el.media.mit.edu/logo-foundation/logo/index.html

Motivated by a desire to introduce opportunities for computational creation in the Computer Clubhouse network of after-school technology programs[2], Scratch builds on the Logo work, as well as the work of others focused on the accessibility of computational creation, including Design By Numbers (Maeda, 2001), LogoBlocks (Begel, 1996), Etoys (Kay, 2005), and Alice (Pausch et al., 1995). Scratch is a visual, blocks-based programming language that enables young people to create interactive media, such as games, stories, and simulations. Scratch's visual approach to programming differentiates it from most programming languages, which typically require the programmer to type text-based instructions (e.g., C, Java, Logo) and grapple with any resulting syntax errors. Young designers snap together Scratch programming blocks to create artifacts in the digital world, just as one might snap together LEGO bricks to create artifacts in the physical world.

The Scratch programming language offers more than 100 programming blocks, grouped into eight different categories (*motion*, *looks*, *sound*, *pen*, *control*, *sensing*, *operators*, and *variables*). In a Scratch project, blocks are used to manipulate the attributes of objects, called *sprites*. For example, blocks in the *motion* category can be used to modify a sprite's movement and position, and blocks in the *looks* category can be used to modify a sprite's visual appearance.

Scratch projects are created by adding sprites and then programming their behaviors by snapping together blocks from these different categories. A sample Scratch project is shown in Figure 2.1. In this project, *control*, *motion*, *looks*, and *sound* blocks have been snapped together, so that when the space key is pressed, a cat (the default sprite) repeatedly dances back and forth to a drum beat, while simultaneously changing color.

Using this basic mechanism of adding sprites and then connecting blocks to specify the behavior of the sprites, kids can develop a wide

2 To learn more about the initial conceptualization of Scratch, I recommend reading the National Science Foundation proposal that funded the initial Scratch development work: http://web.media.mit.edu/~mres/papers/scratch-proposal.pdf

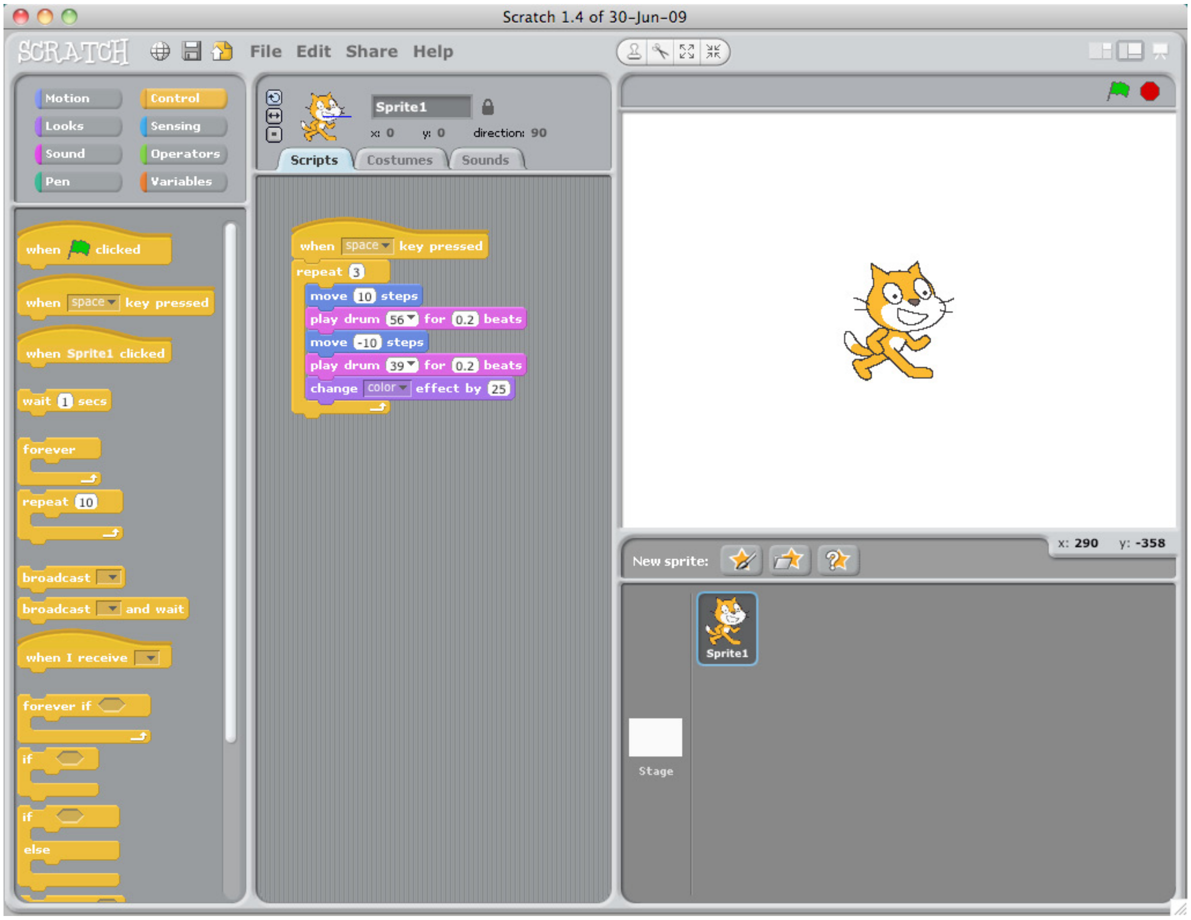


Figure 2.1
Using the Scratch authoring environment to program a cat sprite.

range of personalized projects. Figure 2.2 illustrates some of the possibilities: an interactive joke project, featuring a series of knock-knock exchanges between a monkey and a lion; an interactive art gallery project, featuring explorations into computer-based art; a Sims-inspired project, enabling the user to design their own office; a side-scrolling maze game in which a small green square is navigated past a series of hazards.

In addition to providing an authoring environment for computational media, Scratch provides a setting in which designing, creating, and learning can be shared with others. Papert (1980) described his vision for what such a setting might look like, inspired by a visit to Brazil and to the Brazilian samba schools, the venues in which citizens prepare their Carnival dance performances. He explained what he identified as the essential elements of samba school learning.

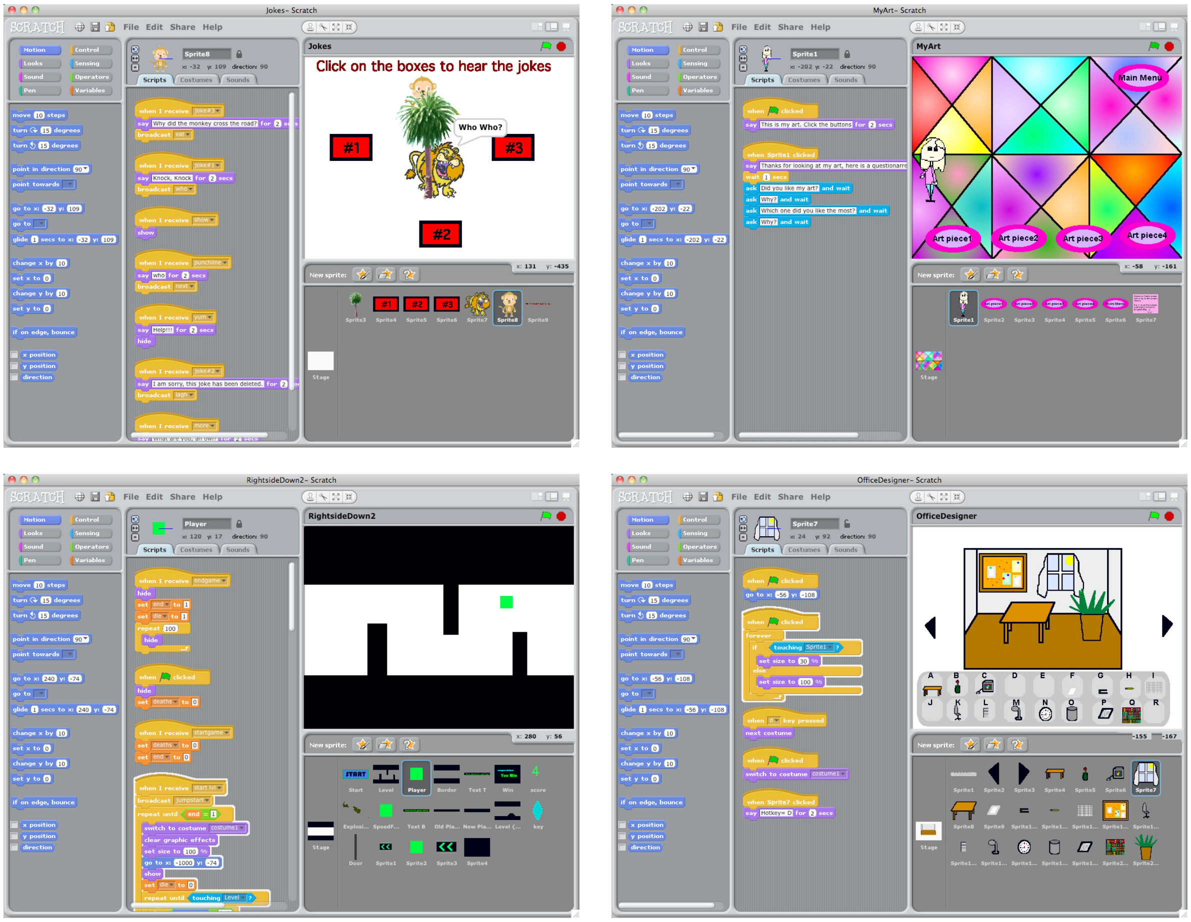


Figure 2.2
(Clockwise, from upper left) Projects that explore a range of genres – stories, art, simulations, and games.

> The samba school, although not "exportable" to an alien culture, represents a set of attributes a learning environment should and could have. Learning is not separate from reality. The samba school has a purpose, and learning is integrated in the school for this purpose. Novice is not separated from expert and the experts are also learning. LOGO environments are like samba schools in some ways, unlike them in other ways. (p. 179)

Although he felt that the samba school vision had not yet been attained with Logo, he was optimistic that samba schools for computational creation would soon be realized.

> I have no doubt that in the next few years we shall see the formation of some computational environments that deserve to be called "samba schools for computation." (p. 182)

Figure 2.3
The Scratch online community homepage.

Inspired by Papert's vision for computational samba schools and by early experiments in communities for computational creation (e.g., Bruckman, 1997), the Lifelong Kindergarten research group created a website to accompany the Scratch authoring environment. This website, *the Scratch online community*, is intended to provide a venue for Scratch designers to connect with one another, sharing their design work and supporting each other's learning (Figure 2.3). Launched in May 2007, the Scratch online community has grown considerably, with hundreds of thousands of members sharing Scratch projects (Resnick et al., 2009). Each day, members, mostly between the ages of 8 and 16, upload more than 2000 new Scratch

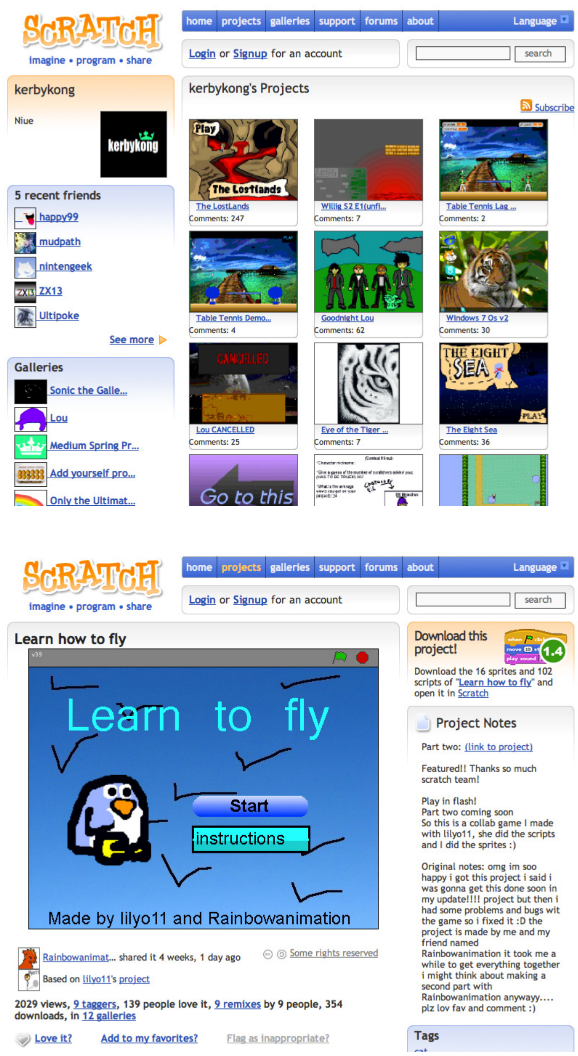


Figure 2.4
Example of Scratch online community (a) profile page, and (b) project page.

projects to the website – and more than 2.8 million projects have been shared in the first five years.

The online community site includes social-networking components, supporting young creators' interactions with one another (Monroy-Hernandez, 2012). Each member has a profile page that displays their projects and other dimensions of their participation, such as their friends and galleries of projects (Figure 2.4a). Members can leave comments on projects, express appreciation for projects through the *Love It* link, and add projects to galleries (Figure 2.4b). In addition to interacting with and providing feedback for others' work, members can download projects to study how they were made, looking at the sprites and blocks of the project. A culture of remixing others' work (with appropriate credit) is strongly encouraged in the Scratch community.

Beyond the help that accompanies the authoring environment (such as help screens and starter projects), the Scratch Team established several sources of help and guidance on the site for newcomers and more experienced creators. Young creators ask and answer questions in the Scratch online community discussion forums (Figure 2.5a). They can find answers to commonly asked questions on various information pages (Figure 2.5b), and find getting-started resources, such as video tutorials and print guides (Figure 2.5c).

I joined the Lifelong Kindergarten research group shortly after the Scratch online community was launched in 2007. It was a fortuitous time to arrive, affording the opportunity to study participation in the online community as it has grown from tens of members to hundreds of thousands of members. (I discuss member demographics and participation data in more detail in subsequent chapters.) The Scratch online community has been my main entry-point for connecting with young people who are working with Scratch in out-of-school settings, primarily at home.

Figure 2.5
Sources of support established by the Scratch Team in the online community: (a) discussion forums, (b) information pages, like the Scratch FAQ, and (c) resources, like the Scratch Cards.

SCRATCHED

Much of the early use of Scratch took place in homes and after-school settings, and many of the young people I interacted with in the early days of the Scratch online community came from home environments that encourage and support creative explorations with technology. But a growing number of schools have started to include Scratch in classroom activities. The adoption of activities like Scratch in schools is essential for broadening and diversifying the community of young people who are participating as computational creators, moving beyond early adopters.

Although the Scratch online community has a large and active membership, it was not designed to support educators – it was designed for people who want to create and share projects, while educators are primarily concerned with helping *other* people create projects. The disparity between the design and teachers' requirements came increasingly into focus when, shortly after joining the group in 2007, I began to receive numerous emails from teachers. In these messages, teachers were sharing stories about their experiences with Scratch. They were requesting curricular resources – or offering to contribute resources that they had created. They were asking questions and offering to respond to others' questions. They were looking for ways to connect with other educators working with Scratch who were nearby or had similar interests.

Based on these educator interests and motivated by the *community of practice* literature – a model in which teachers as learners have access to peers, shared goals, and resources (Wenger, 1998; Barab, Barnett, & Squire, 2002) – I developed the *ScratchEd* site for educators. Teachers interested in or already actively working with Scratch can use ScratchEd to share stories, exchange resources, ask and answer questions, and find other educators (Figure 2.6). In designing the ScratchEd site, I was inspired and influenced by others' work in online communities for educators, including Tapped In (Farooq, Schank, Harris, Fusco, & Schlager, 2007), KNOW (Brunvand, Fishman, & Marx, 2005), WIDE World (Wiske, Perkins, & Spicer, 2006), and Inquiry Learning Forum (Barab, MaKinster, & Scheckler, 2003).

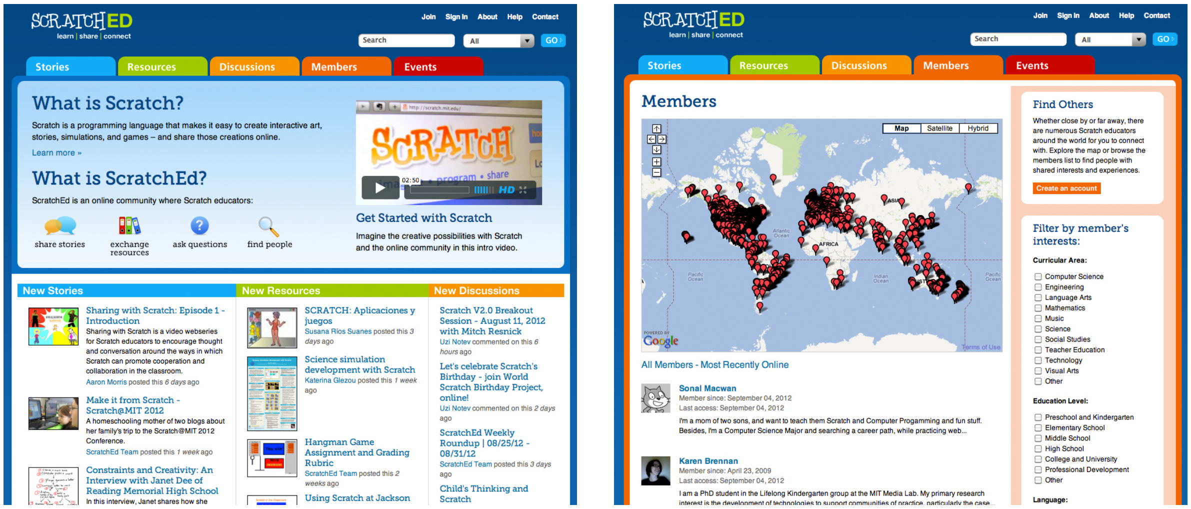


Figure 2.6
The ScratchEd online community front page and Members page.

ScratchEd made its public debut in August 2009. Since then, in its first three years, more than 5800 educators from around the world have joined the community, and have contributed more than 180 stories, 470 resources, and 2,800 discussion posts. Over the past year, the site has received an average of 62,000 page views from 11,000 unique visitors per month, predominantly from the United States.

Although some of the initial teachers working with Scratch employed learner-centered, agency-supporting approaches in their teaching, many others adopted more traditional, teacher-centered, "instructionist" strategies due to various pressures, such as insufficient support, lack of resources, expectations about roles, or challenges in accommodating standards (Rainio, 2008; Sawyer, 2004). With these constraints, the structure of school often operates in opposition to a learner-agency-based approach, for example, by enforcing homogeneity in activity or relying on models of learning as an individual activity.

But teachers have a crucial role to play in supporting learner agency within learning environments. As discussed in the previous chapter, supporting learner agency does not imply removing all structure and support. Teachers can provide much-needed metacognitive support, helping students define problems, persist through challenges, and reflect on experiences – with the teacher modeling to the learners what being a learner can look like (Brown, 1994; Duffy

& Cunningham, 1996; Hmelo-Silver, 2004; Scardamalia & Bereiter, 1991).

Beyond pedagogical challenges, teachers also often lack comfort and confidence with technology. More than a decade has passed since Schofield (1995) and Cuban (2001) wrote about the challenges of meaningfully introducing technology in K-12 classrooms, yet many of the same problems – including lack of understanding about how to include technology and lack of support to improve that understanding – persist (Buckingham & Willet, 2006; Buckingham, 2007; Palfrey & Gasser, 2008; U.S. DOE, 2010).

While many models have been proposed to support teachers, particularly in supporting teachers' experimentation with new pedagogical strategies and with use of technology, collaboration among teachers has proven to be particularly effective (Dexter, Anderson, & Ronnkvist, 2002; Dexter, Seashore, & Anderson, 2002; Fuller, 2000; Schlager & Fusco, 2003). A blend of online and face-to-face interactions best supports a community of practice, with online interactions and face-to-face interactions mutually reinforcing the development of relationships, understanding of practice, and building of capacity among teachers (De Souza & Preece, 2004; Goodfellow, 2005; Hew & Hara, 2007; Kirschner & Lai, 2007; Vaughan, 2004).

Based on this research, it was evident that, although the ScratchEd online community provided some support for teachers' understandings of Scratch and abilities to develop student-centered Scratch learning environments, additional support was needed. Support was also needed to better accommodate all of the settings in which Scratch is being used: across disciplines, from computing studies to language arts to science to visual arts, across ages, from kindergarten to college, and by educators who have varying levels of familiarity with Scratch and computational creation.

Accordingly, we have been expanding the ScratchEd professional development offerings, work made possible by funding from the National Science Foundation Discovery Research K-12 program. First, we have been featuring stories, resources, and discussions in the ScratchEd online community that highlight various strategies and approaches for designing learning environments. Second, to accompany the ScratchEd online community activities, we have been organizing face-to-face and online gatherings where teachers can develop deeper understandings of Scratch and student-centered approaches to learning. This has included quarterly introductory workshops for educators new to Scratch, monthly meetups for educators with some Scratch experiences, and monthly webinars that are recorded and shared on ScratchEd.

Finally, we have been developing resources for teachers to use when introducing Scratch to students and when conducting workshops for their colleagues. For example, I wrote a curriculum guide for Scratch that was released in September 2011. The guide was downloaded more than 34,000 times and translated into more than 7 different languages (including Portuguese, Spanish, Turkish, Korean, and Traditional Chinese) by members of the ScratchEd community in the year following its release. Accessing and exploring these resources has been made easier by connecting announcements to other channels, such as email, Twitter, and Facebook.

It is through the ScratchEd online community and gatherings that I have been able to learn more about the experiences of educators. This community has served as my main entry-point for connecting with teachers who are working with Scratch in schools, primarily K-12 classrooms.

CONSTRUCTIONIST ASPIRATIONS

This work focuses on the aspirations and experiences of kids and teachers working with Scratch – kids who are having learning experiences at home, in the Scratch online community, and teachers who are creating learning experiences for their students, in K-12 classrooms – and how various structures enable and constrain those aspirations.

But my explorations of kids' and teachers' perspectives are necessarily shaped by my own assumptions and goals: if Scratch can be used in any way, how do *I* hope it will be used? My aspirations for Scratch are rooted in assumptions about what "good" learning is – and I use the rest of this chapter to describe these underlying assumptions.

My assumptions are connected to the constructionist tradition – an approach to learning that emphasizes the importance of constructing, building, making, and designing as ways of knowing. This tradition, in turn, builds on constructivist arguments that learning does not happen through a process of transfer or acquisition. As Kafai and Resnick (1996) described, "knowledge is not simply transmitted from teacher to student, but actively constructed by the mind of the learner. Children don't get ideas; they make ideas" (p. 1). A learner constructs new models and understandings that are connected to the learner's existing structures and models (Duffy & Cunningham, 1996; Scardamalia & Bereiter, 1991).

Constructionism is grounded in the belief that the most effective learning experiences grow out of the active construction of all types of things, particularly things that are personally or socially meaningful (Bruckman, 2006; Papert, 1980), that are developed through interactions with others (Papert, 1980; Rogoff, 1994), and that support thinking about one's own thinking (Kolodner et al., 2003; Papert, 1980). These four aspects of constructionism – which I define here as learning through the activities of *designing*, *personalizing*, *sharing*, and *reflecting* – are key activities that, I argue, should be included in the design of learning environments.

These constructionist activities are not only beneficial for young people participating as designers of interactive media with Scratch, but also for teachers in their roles as learners in professional development (PD) activities. Many PD opportunities treat teachers as consumers, neglecting fundamental understandings about how people learn, as evidenced by language like "teacher training". As Papert (1993) argued,

> Although the name is not what is most important about this concept, it is curious that the phrase "teacher training" comes trippingly off the tongues of people who would be horrified

> at the suggestion that teachers are being trained to "train" children. (p. 70)

The constructionist principle that "learning by doing is better than learning by being told" (Bruckman & Resnick, 1995, p. 94) applies equally well to teachers as to students, and supports teachers' understandings of the types of experiences desired for their students. As such, my approach in the ScratchEd PD activities has been to create opportunities for teachers to engage in the same designing, personalizing, sharing, and reflecting activities that are essential for young people.

With an emphasis on creating and agency, constructionist approaches to learning are well aligned to the demands and expectations of computational culture and I now describe the theoretical contexts of the four activities – designing, personalizing, sharing, and reflecting – that I define as central to a constructionist approach. Each of these activities has an extensive literature associated with it; here, I draw attention to a few of the key ideas, themes, theories, and concepts that have been most helpful to my understanding.

Designing

There are competing narratives about the relationships between young people and digital technology. One narrative is that of the "digital native" – kids who were "born digital" and belong to the "digital generation" (Palfrey & Gasser, 2008; Prensky, 2001; Tapscott, 2008). This narrative is often centered on an assumed familiarity and fluency with computation, that young people have innate understandings that elude adults – parents and teachers, cast as "digital immigrants".

Descriptions of digital natives' activities and participation, such as Jenkins et al.'s (2006) *core media literacy skills*, Ito et al.'s (2009) *hanging out, messing around, and geeking out participation modes*, and others that similarly draw on exemplars or ideal types, have elicited criticism for misrepresenting the "often unspectacular" interactions between young people and computational culture (Selwyn, 2009, p. 364). Some versions of the "digital native" narrative tend toward an exaggerated or undifferentiated view of technology use – that all forms of interaction with digital technologies are valuable and all types of participation offer equally interesting opportunities for learning. Buckingham (2007) provided a broad critique of the young-person-as-technology-elite narrative, arguing that the narrative is less of an observation than an aspiration.

> Recent studies suggest that most young people's everyday uses of the Internet are characterized not by spectacular forms of innovation and creativity, but by relatively mundane forms of communication and information retrieval. … The discourse of the "digital generation" is precisely an attempt to construct the object of which it purports to speak. It represents not a description of what children or young people actually are, but a set of imperatives about what they should be or what they need to become. (p. 14)

The creative activities of designing and making with digital technologies are underrepresented in the activities of young people. This is partly due to the nature of the technologies themselves, for example, the preponderance of "edutainment" software, and the paucity of construction-oriented software (Ito, 2009). But it is also partly due to the lack of visibility and value in school culture (and beyond) of design thinking, with young people reluctant to see the complexities of design activities "as opportunities rather than as things to be avoided" (Fischer, 2002).

Despite the lack of opportunities and visibility, however, this is an important moment in history for design – both in the context of digital technologies and beyond – and the habits that are cultivated through designing (Burdick & Willis, 2011). Consider the call to action in the 2010 National Education Technology Plan –

> Whether the domain is English language arts, mathematics, sciences, social studies, history, art, or music, 21st-century competencies and such expertise as critical thinking, complex problem solving, collaboration, and multimedia

> communication should be woven into all content areas. These competencies are necessary to become expert learners, which we all must be if we are to adapt to our rapidly changing world over the course of our lives. (p. 9)

Design activities respond to this call, by engaging young people in iterative thinking, problem-solving practices, and critical creativity, which serve as the foundation of learning (Harel & Papert, 1990; Kafai, 1995; Kolodner et al., 2003; Krajcik & Blumenfeld, 2006). Designing necessitates the ability to identify and negotiate constraints, clarify and manage ambiguity, and, fundamentally, persist and engage in hard work (Fischer & Nakakoji, 1997; Razzouk & Shute, 2012; Sawyer, 2006; Seiter, 2008).

Personalizing

Despite what the structures of modern education, such as large class sizes and homogeneous curriculum, might suggest, the individual matters. Personalizing, as a constructionist aim, means that the design of learning experiences should consider how to engage an individual learner on multiple levels, including cognitive and affective.

The cognitive perspective on personalizing traces back to constructionism's main influence – Piaget and constructivist assumptions about learning. In constructivist theories of learning, learning is not something done *to* learners, but rather something done *by* learners. Learners are not filled with knowledge and new ideas by the world around them; they engage in processes of adaptation. Engaging with new ideas leads to *assimilation*, by taking the new ideas and connecting them to already-established understandings – or to *accommodation*, by modifying already-established understandings in consideration of the new ideas (Ackermann, 1996; Koschmann, Kelson, Feltovich, & Barrows, 1996; Piaget, 1929). Understanding and supporting learning necessarily means creating opportunities to make sense of the individual, personal connections that learners are forming to what they are learning.

Part of this sense-making involves thinking about differences in individuals' learning styles and self-concepts, and recognizing that there is not one way or style of learning. There are numerous examples of frameworks that seek to extend the ways in which learners see themselves and are seen by others. Gardner's multiple intelligences (1983, 1991, 1999) aimed to dislodge some of the privilege associated with linguistic and logical/mathematical capacities, by drawing attention to other capacities, such as musical, spatial, and inter/intrapersonal. Gilligan's reinterpretation of Kohlberg's stages of moral development (1982) sought to displace masculinist assumptions about self versus other. Dweck's (2000) entity and incremental theories of intelligence provided ways of thinking about how to productively support students, by challenging assumptions about ability, success, praise, and confidence. Turkle and Papert (1990), in critiquing Piaget and Inhelder's privileging of formal reasoning, argued for recognition of both bricoleur and planner approaches, particularly in the planner-dominated culture of computation. These frameworks deserve the attention of learning environment designers, and should encourage thinking about how individual learners are more or less productively engaged by different strategies.

With personalizing, there is always a tension between the inside/outside, in relation to the learner. With assimilation/accommodation, tensions arise between the ideas that are being encountered and the ideas already in place. With learning styles, tensions arise between an individual's learning style and the external supports (whether a teacher or a technology or some other aspect of the learning environment) available to the individual learner. As a final example of the inside/outside forces at work in personalization, tensions arise between intrinsic and extrinsic motivation.

Perkins (1986) described intrinsic motivation from the perspective of a classroom observer, noting that different learners demonstrate different levels of enthusiasm and interest.

> Psychologists have taken an interest in interest and have developed a way of talking about enthusiasm – "intrinsic motivation." When you have intrinsic motivation for an

> activity, you value the activity for its own sake. You enjoy doing it, find it challenging but not too frustrating, and view it as worthwhile in itself. (p. 116)

Intrinsic motivation is an important component for supporting creativity, successfully dealing with challenges, and deeply engaging in one's interests (Csikszentmihalyi, 1990; Perkins, 1986; Sawyer, 2006). Yet it is easily undermined by external factors, such as badges and grades and praise (among others), which intentionally or unintentionally control learners as forms of extrinsic motivation (Dweck, 2000; Hennessey & Amabile, 1998; Kohn, 1999; Tough, 2012).

As learning increasingly takes place outside of school settings, driven predominantly by intrinsic motivation and learners' desires to pursue their interests, questions emerge about how to cultivate and amplify intrinsic motivation both within and beyond the classroom setting. A recurrent theme in the literature is the need to balance challenge and current ability (Csikszentmihalyi, 1990; Dweck, 2000; Perkins, 1986). Too much challenge leaves learners feeling frustrated; too little challenge leaves learners feeling bored. Neither situation is desirable, particularly if it persists.

Sharing

Learning and development have important individual components (as articulated in *personalizing*, from the perspective of Piaget's work), but they are also deeply social processes. Vygotsky extended the Piagetian framing of the individual's cognitive processes by introducing the notion of the *zone of proximal development* (ZPD), defined as "the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers" (1978, p. 86). Vygotsky's notion of the ZPD expanded the boundaries of individual cognition, including other people and their abilities as part of an individual's capacities for taking on challenges of increasing difficulty (Cole & Wertsch, 1996).

Theories about communities of practice and situated learning further extend thinking about how others support learning, in particular, how community settings can provide access to other learners and artifacts (Brown, Collins, & Duguid, 1989; Lave & Wenger, 1991; Rogoff, 1994). In this literature, apprenticeship is a recurring metaphor for the type of learning that can take place, introducing new ways of thinking about the learner and the people around the learner who are helping them (Collins, 2006; Lave & Wenger, 1991; Wenger, McDermott, & Snyder, 2002). Learners are gradually folded into relationships with other learners, understandings of the enterprise of the learning, and familiarity with the objects and practices of the community – learning from those with greater experience and expertise, in a process that Lave and Wenger (1991) described as *legitimate peripheral participation*.

> "Legitimate peripheral participation" provides a way to speak about the relations between newcomers and old-timers, and about activities, identities, artifacts, and communities of knowledge and practice. It concerns the process by which newcomers become part of a community of practice. A person's intentions to learn are engaged and the meaning of learning is configured through the process of becoming a full participant in a sociocultural practice. This social process includes, indeed it subsumes, the learning of knowledgeable skills. (p. 29)

More recent research has described the ways in which the social nature of learning serves as essential motivation and support for young people's participation in computational culture, particularly in the context of online interactions (Buckingham & Willett, 2006; Ito et al., 2009; Jenkins et al., 2006). Whether hanging out with friends or playing games or remixing media, having access to others makes for better participation, as young people are able to support each other in understanding practices and norms. Bruckman's (1998, 2006) work described the cognitive, social, and psychological benefits that an online community provided for individual learners in constructionist activities. From technical support to emotional support, having access to others bolstered individuals' capacities for creative work. And, as described earlier in this chapter, the social nature of learning is not reserved for kids – teachers as learners can similarly benefit from access to others (Fishman & Davis, 2006).

Reflecting

In Mindstorms, Papert described his vision of children as epistemologists, wherein kids use computers as an opportunity to explore their processes of thinking. Programming becomes a context for thinking about thinking, and the LOGO programming language serves as something to think with.

The activities of designing, personalizing, and sharing invite learners to ask numerous questions of themselves, of what they are doing, and of how they are thinking. *What do I want to create? What do I need to create it? What do I need help with? Why didn't that work as I expected it to? Who might help me? Who might I help? How might I better approach all of these questions?* These types of questions represent opportunities for kids to reflect on their activities and to think about their thinking – for kids to engage in *metacognitive* processes.

Numerous frameworks have been proposed for articulating metacognitive processes, and several highlight the temporal dimension of metacognition – *when* the thinking about thinking takes place in relation to action. Schön (1983) articulated a difference between reflection-in-action and reflection-on-action. Bransford et al. (2000) emphasized a similar separation, drawing out self-regulation and reflection from metacognition, with the former focusing on activity planning and monitoring, and the latter focusing on assessment and evaluation of activity performance. Flavell (1979) described metacognition as the interplay between goals (what the learner is trying to achieve), strategies (how the learner tries to achieve it), metacognitive knowledge (what the learner knows about learning), and metacognitive experiences (how the learner thinks about that knowledge in action).

The significance of metacognition in a variety of learning and cognitive processes has long been recognized.

> Metacognition plays an important role in oral communication of information, oral persuasion, oral comprehension, reading comprehension, writing, language acquisition, attention, memory, problem solving, social cognition, and, various types of self-control and self-instruction. (Flavell, 1979, p. 906)

The ideas of self-control and self-instruction described by Flavell – varyingly referred to as self-control, self-instruction, self-regulation, self-efficacy, self-directedness – speak directly to the idea of learner agency. Bandura (1997) highlighted the significance of these capacities, for supporting learning as both a life-long and life-wide activity.

> Development of capabilities for self-directedness enables individuals not only to continue their intellectual growth beyond their formal education but to advance the nature and quality of their life pursuits. Changing realities are placing a premium on the capability for self-directed learning throughout the life span. The rapid pace of technological change and the accelerated growth of knowledge require continual upgrading of competencies if people are to survive and prosper under increasingly competitive conditions. (p. 227)

Constructionism, Agency, And Structure

As described in the previous chapter, the tension between agency and structure is a central concern in the design of learning environments – particularly in constructionist approaches, which, unlike some other models of learning, emphasize learner agency.

In contrast, consider the "programmed instruction" model developed in the mid-20th century by influential behaviorist-psychologist B.F. Skinner. In this model, learners work with "teaching machines" that lead them through a series of content questions, for example, math problems or spelling challenges. The problems have specific answers and are introduced in an intentional, guided sequence. In a 1986 Phi Delta Kappan article, Skinner responded to critics who argued that the model was too controlling and gave students little (if any) freedom in defining and grappling with problems themselves.

> Some 350 years ago Comenius said, "The more the teacher teaches the *less* the student learns," but that is true only if it means "the less the student learns about learning." (p. 109)

His rebuttal highlights a fundamental difference between behaviorist and constructionist values and methods. While Skinner dismissed the importance of students "learning about learning", it is of particular value in a constructionist approach.

When confronted by low-agency/high-structure approaches, such as Skinner's programmed instruction method, the reaction of some constructionists has been to *reject* structure, usually framed as a rejection of instruction in favor of pure discovery (as described in the previous chapter). But this need not be the case. As Kafai (2006) noted, "a common myth associated with constructionism is the idea that all instruction is bad" (p. 36). Constructionism does not inherently reject structure; rather, it invites teachers and learners to challenge the assumptions about the conditions and structures necessary for learning, and to carefully consider the complexity of the culture of learning. As Papert (1993) observed,

> Even the statement (endorsed if not originated by Piaget) that every act of teaching deprives the child of an opportunity for discovery is not a categorical imperative against teaching, but a paradoxically expressed reminder to keep it in check. (p. 139)

Teaching and instruction are only one form of structure – in the rest of this work I explore the cultures of learning outside and inside school, the different structures of these settings, and how the structures of these two settings enable (and constrain) learner agency.

Chapter 3

CONVERSATIONS

In this chapter, I describe the data collection and analysis procedures that I employed to develop understandings and descriptions of computational creation in out-of-school and in-school learning environments.

This work is focused on developing understandings of how the structures of learning environments support (or undermine) agency in computational creation, using Scratch out of school and Scratch in school as particular settings for computational creation.

My methodological approach is primarily qualitative, as a way of attending to the culture, community, and context of these learning environments, as well as the lived experiences of individual young people and teachers. Stake (2010) described this dual focus – and process/representation – of qualitative research.

> It is common for people to suppose that qualitative research is marked by rich description of personal action and complex environment, and it is, but the qualitative approach is equally distinguished for the integrity of its thinking. There is no one way of qualitative thinking, but a grand collection of ways: It is interpretive, experience based, situational, and personalistic. Each researcher will do it differently, but almost all of them will work hard at interpretation. They will try to convey some of the story in experiential terms. They will show the complexity of the background, and they will treat individuals as unique, yet in ways similar to other individuals. (p. 31)

I draw inspiration from ethnography (LeCompte & Schensul, 1999; Van Maanen, 1988), case study (Yin, 2009), and design-based research (Anderson & Shattuck, 2012; The Design-Based Research Collective, 2003). All three traditions acknowledge the importance of real-world (as opposed to controlled, laboratory) experiences – and the corresponding complexity of these settings and processes, which require detailed descriptions and cannot be reduced to a simple set of variables or factors. Detailed descriptions invite dense data collection (including interview, observation, artifact analysis, database analysis, and survey), and the use of multiple types and instances of data serve to minimize bias in interpretation during analysis.

In this chapter, I describe my process for developing understandings and descriptions of the out-of-school and in-school learning environments. For each setting, my primary source of data is conversations with the people primarily responsible for navigating and negotiating the various structures of the learning environment. For the out-of-school setting, I focus on conversations with kids working with Scratch at home, through the Scratch online community. For the in-school setting, I focus on conversations with teachers working with Scratch at school, in K-12 classrooms.

The chapter is organized into two parts. The first part – *Collecting* – explains the data collection procedure, describes the backgrounds of the people I spoke with and their representativeness, and discusses the de/limitations of the data collection process for each of the two settings. The second part – *Analyzing* – explains my analytical approach, describing how the theoretical framework of agency and structure serves as a foundation for subsequent analysis.

COLLECTING

The Scratch online community and (to a lesser extent) the ScratchEd online community have provoked excitement among some researchers. The excitement derives from a sense that these online communities and the interaction data that they are collecting possess the answers to important questions about learning – a sense that, as one researcher exclaimed to me, "We can know everything that kids and teachers are doing!" This sentiment is, as boyd and Crawford (2012) argued, part of the "mythology" of "Big Data" – "the widespread belief that large data sets offer a higher form of intelligence and knowledge that can generate insights that were previously impossible, with the aura of truth, objectivity, and accuracy" (p. 2).

I do not deny the excitement and potential interestingness of the large data sets afforded by sites such as the Scratch and ScratchEd communities, but I share the sentiment of caution expressed by boyd, Crawford, and others. There is a complexity in intention behind action that is challenging to capture with data logging. If we use kids interacting in the Scratch online community as a concrete example of this complexity, there is a type of simultaneous depth and superficiality to what can be said about what they do. We can know across all members – or for a particular member – how many times they have remixed a project. But why are they doing it? How does what they *are* doing relate to what they *want* to do? What are they choosing not to do? What are they unable to do? How are they thinking about what they (and others) are doing? As Manovich (2012) reminded, although there are many things to learn and understand from analysis of large data sets, "we just have to keep in mind that all these data are not a transparent window into peoples' imaginations, intentions, motives, opinions, and ideas" (p. 466).

These details of participation and motivation are needed to understand how kids and teachers are making sense of and employing Scratch in out-of-school and in-school learning environments. Consequently, conversations with kids and teachers (which offer a different type of depth than mining the large data sets of the online communities) serve as the central component of my data collection for both settings. Conversations, in the form of semi-structured interviews with young Scratchers and teachers, are a way "to understand the world from the subjects' points of view, to unfold the meaning of peoples' experiences, to uncover their lived world" (Kvale, 1996, p. 1). I triangulated the conversation data with other sources, which I describe in each of the following sections.

Kids In The Online Community

I have been observing young people's participation in the Scratch online community for the past five years, drawing on a key (virtual) ethnographic principle of "learning through immersion" (Hine, 2008, p. 259). Based on this observation work, I have been writing field notes and memos about Scratchers' activities, and saving artifacts of their work – primarily Scratch projects, but also other electronic artifacts (such as forum posts, emails, and blog entries).

Observation helps make sense of *what people do*, but provides limited insight into *how people think* about their actions and behavior. As such, I have also been conducting (with support from ScratchEd research team members) in-depth interviews with kids in the online community. This approach – trying to understand the culture of computational creators from the perspective of computational creators – is particularly important, as a way of supporting learner agency through learner voice. An important goal of the interviews was for kids to have the opportunity to tell stories about their experiences, as "stories lived and told educate the self and others" (Clandinin & Connelly, 2000, p. xxvi).

The interviews were semi-structured and were approximately 60-120 minutes in duration. The interview questions, listed below, were organized into four major sections (with sample questions listed in italics).

1. Background
   a. Introduction to Scratch: *How did you find out about Scratch? What is Scratch?*
   b. Current practices: *Where do you use Scratch? What do you do with it? Do other people help you? Do you help other people?*
2. Project creation
   a. Project framing: *How did you get the idea for your project (one of several projects that are discussed)?*
   b. Project process: *How did you get started making your project? What happened when you got stuck?*
3. Online community
   a. Introduction to the online community: *What do you do in the online community? What is the Scratch online community?*
   b. Other people, other projects: *How do you find interesting people and interesting projects? How do you interact with other Scratchers?*
4. Looking forward
   a. Scratch: *What do you dis/like about Scratch? What would you keep, add, change?*
   b. Technology: *What are other tech-related things you like to do?*
   c. Beyond technology: *What are other non-tech-related things you like to do?*

30 kids from the Scratch online community were interviewed between November 2007 and January 2012. Each interview was conducted by a pair of researchers (typically, myself and one of

several graduate-student interns from the Harvard Graduate School of Education). 27 interviews were conducted via phone or Skype and three interviews were conducted in person. The interviewee selection process combined random sampling and open recruitment. For the random sampling component, I segmented the online community into four groups: group A (kids between the ages of 7 and 18 who had joined the community within the past four weeks, and had not extensively participated in the community, either socially or technically – roughly Scratch "novices"), group B (kids between the ages of 7 and 18 who had been a part of the community for more than four weeks, and had moderate levels of social and/or technical participation), group C (kids between the ages of 7 and 18 who had been a part of the community for more than four weeks, and had high levels of social and/or technical participation), and group D (kids between the ages of 7 and 18 who had been a part of the community for more than four weeks, had high levels of social and/or technical participation, and had significant status in the community).

Using these categories, I ran database queries to generate lists of usernames for each category, and then randomly selected a subset of these community members to invite for conversations. There were several challenges with this approach, particularly with attaining adequate representation of group A and group D kids.

With group A invitations, I realized that kids who had just joined the community might feel overwhelmed or intimated by a request to talk about their very recent experiences, even though the invitations downplayed the importance of experience or expertise. Another challenge with the group A invitations was false positives – kids who appeared to be novice, but who were actually much more advanced, having created an additional account to, for example, experiment with a different genre of project-making. With group D interviews, I found that "Scratch famous" kids were often negotiating numerous community commitments and many felt that their limited time was better directed toward cultivating the attention of their fans.

In addition to this community sampling, I also posted a general invitation to members of the Scratch online community. This was partly in response to the challenges of the random sampling response rates, but also based on a desire to avoid alienating community members by being perceived as favoring certain members with invitations.

The selected interviewees represent a range of ages (8-17), geographic locations (mostly U.S., but including several European countries and Canada), durations of participation (from a few weeks to almost five years), and technical/aesthetic sophistication (from beginners to sophisticated programmers and designers). 37% of the interviews were conducted with female Scratchers, which is consistent with the proportion of female participation in the online community. All of the interviewees worked with Scratch primarily at home (and, in the majority of cases, exclusively at home).

Table 3.1 presents a demographic overview of the 30 kids, including their interview alias, age, gender, location, how they were introduced to Scratch, and the group (A/B/C/D) to which I assigned them. Group assignment was based on quantitative participation data and qualitative interview data.

In Figure 3.1, I compare the participation of these 30 kids (clustered by group) to the 75,568 Scratch community members (6.2% of 1,222,242) who have posted four or more Scratch projects and have been active in the community for at least 28 days. I argue that the other 93.8% of Scratch accounts (with either less than four projects or less than four weeks of participation) constitute an uninteresting demographic to compare with the interviewees, as these accounts represent, at best, peripheral participation in the online community. For example, it is very easy to create a Scratch account and, comparatively, more difficult to create Scratch projects, so it is unsurprising that 70.6% of all registered members (863,178 of 1,222,242) have not posted a project to the online community.

Appendix A provides a more detailed quantitative portrait of the kids' participation in the online community.

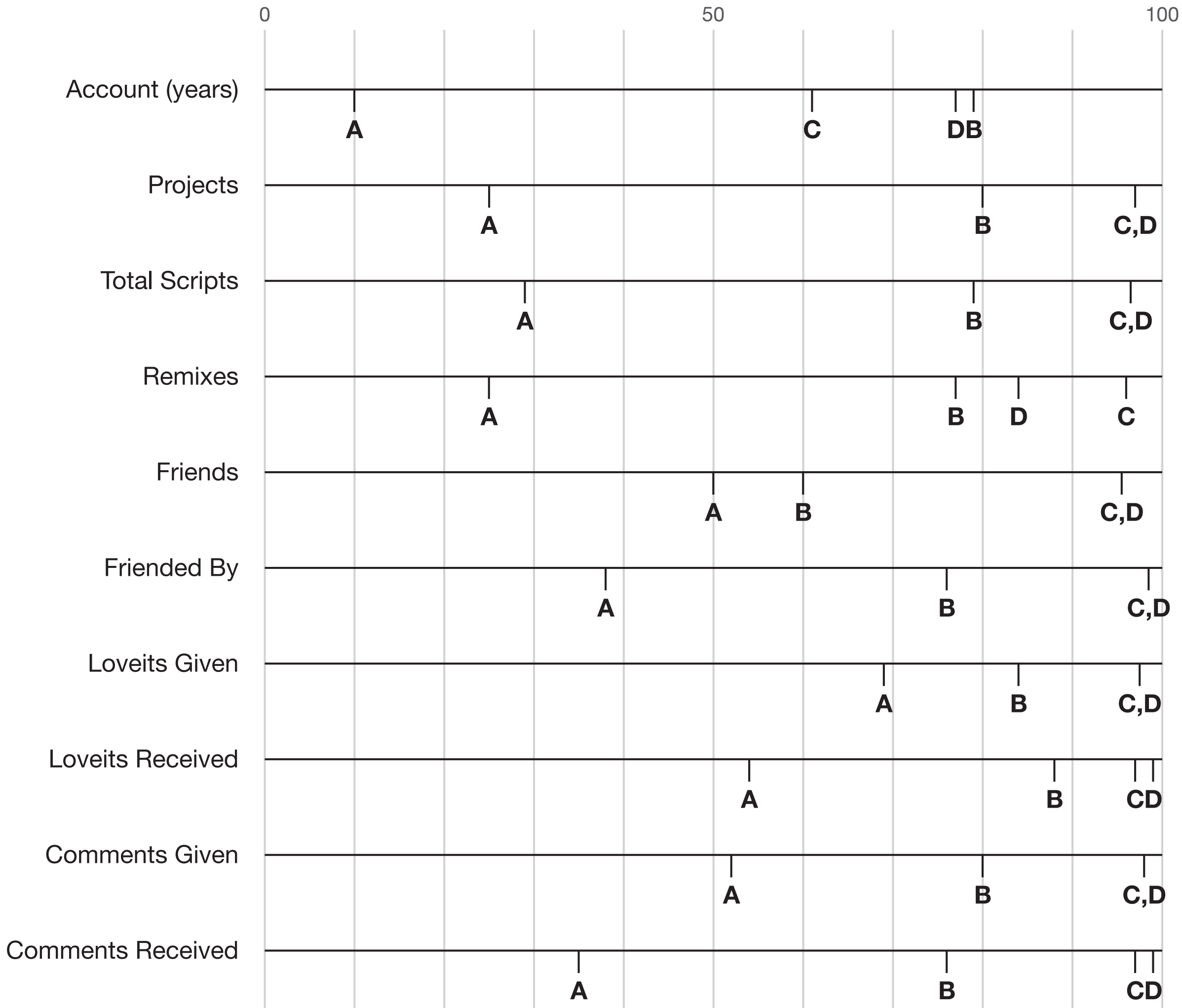


Figure 3.1
Participation in the Scratch online community: interviewee groups compared with the population of active community members. Percentile rank within the community increases along the horizontal axis.

There are limitations and delimitations of the sample and of the comparison. First, as limitations, with the current instrumentation of the site, it is not possible to distinguish kids working at home from kids working in other settings (such as school or informal learning environments like libraries and museums). It is also not possible to determine when multiple accounts belong to a single individual. For example, some kids create two accounts – one account serves as the account for finished work, and a secondary account is used for works-in-progress and testing.

Second, as a delimitation, I am focusing on kids who are working with Scratch at home, but who are working with the Scratch online

Table 3.1 Summary of demographic information for the out-of-school interviewees.

Group	Name	Gender	Location	Age	Introduced to Scratch
A	Allison	F	United States	13	Father
A	Adele	F	United States	9	Teacher Recommendation
A	Robin	F	United States	10	Teacher Recommendation
A	Evan	M	United States	10	Father
B	Jackson	M	United States	11	MIT Website
B	Jenson	M	United States	11	Father
B	Brook	M	United States	8	Aunt
B	Connor	M	United States	12	Teacher Recommendation
B	Easton	M	United States	10	Friend
B	Edgar	M	United States	12	Online Search
B	Brent	M	United States	8	Friend
C	Sebastian	M	United States	13	Teacher Recommendation
C	Aaron	M	United States	10	Community Program
C	Devon	M	United States	10	Mother
C	Barry	M	United States	13	Friend's Father
C	Matt	M	United States	14	Aunt
C	Lindsey	F	United States	12	Teacher Recommendation
C	Ashleigh	F	United States	14	Teacher Recommendation
C	Nevin	F	Australia	9	Friend
C	Fletcher	M	Europe	14	Online Search
C	David	M	United States	9	Father
D	Chuck	M	United States	14	Mother
D	Sonia	F	United States	16	Online Search
D	Chelsey	F	Europe	10	Father
D	Lori	F	United States	14	Father
D	Bradley	M	United States	12	Teacher Recommendation
D	Clark	M	United States	12	Father
D	Jan	M	Europe	16	Online Search
D	Eva	F	Canada	13	Father
D	Lana	F	Europe	17	Uncle

community as an important component of their activities. There are kids working at home who intentionally (or unintentionally) do not participate in the online community. Consider the experiences of home-based Scratchers Simon, Terry, Sam, and June. Simon was introduced to Scratch by his father, who had installed the authoring environment on the family computer, but had not mentioned that there was an accompanying online community. Terry knows about the Scratch online community, but doesn't visit or share his work, as he doesn't want his projects to be "copied". Sam saw the other projects in the online community and felt demoralized, that his projects weren't good enough to be included. June isn't allowed to use the website, as her parents feel that she's too young to be interacting with other kids in a social networking environment.

Teachers In Schools

Designing the ScratchEd professional development model – which includes the online community, face-to-face and virtual gatherings, and resources – over the past five years has provided opportunities to study how teachers make sense of computational creation (in the context of Scratch) and of learner-centered approaches to designing learning environments. The stories, resources, and discussions that educators have posted to the ScratchEd online community have been rich sources of data about teaching practice – what people describe doing and the resources that they are creating or seeking in support of their goals. Observations of Scratch educator meetups, workshops, and classroom visits have also provided an important source of data for understanding educators' teaching practices.

These observations and artifact analyses serve to complement the primary source of data – interviews with educators working with Scratch in K-12 school settings. The interview protocol evolved over time to highlight different areas of focus, but was generally organized into three major sections: previous teaching and technology experiences, approach to including Scratch in the classroom, and motivations and reflections on using Scratch in teaching practice. A recent iteration of the protocol is listed below (with sample questions listed in italics).

1. Background
    a. Current position: *What is your current position? How long have you been in your current position?*
    b. Teaching and technical background: *How long have you been teaching? What types of teaching experiences have you had (e.g., ages/grades, in/formal, disciplines)? What did you study in school? What is your technical background, if any? How would you describe your comfort level with technology?*
2. Scratch experience
    a. Scratch background: *How did you hear about Scratch? How long have you been using Scratch?*
    b. Setting: *In what setting are you using Scratch (e.g., # of students, course description, duration of time)? What is the school culture/attitude toward technology? What can you tell me about the students you work with (e.g., age, socioeconomic status, gender)?*
    c. Implementation: *How did you use Scratch? How were the classroom activities designed? How do you think about your role? How do you think about your students' role? What resources did you use? How do you assess students?*
3. Reflections
    a. Motivation: *Why are you interested in Scratch? What do you think Scratch is good for? What is the value of Scratch to students, from your perspective as a teacher and from a student's perspective?*
    b. Success: *What does a successful experience with Scratch look like? In what ways was your experience successful or not? What does a teacher need in order to be successful with Scratch?*

30 interviews with Scratch educators were conducted between October 2009 and May 2012. Each interview was conducted via phone or Skype by a pair of researchers (typically, myself and one of several graduate-student interns from the Harvard Graduate School of Education). The educators were invited to participate and represented a range of grades (from K to 12), teaching experience (from less than one year to more than 30 years), settings (required courses, elective courses, in-school, after-school), subject areas (cross-curricular, IT studies, computer science), and roles (classroom teachers, technology coaches, mentors, teacher educators). The educators

Table 3.2 Summary of demographic information for the in-school interviewees.

Name	Gender	Position	Years Teaching	Technical Background	Introduced to Scratch	Years w/ Scratch
Clare	F	After-School Mentor	5	Formal	MIT Media Lab	3
Selena	F	Teacher Educator	5	Formal	MIT Media Lab	3
Sun	M	Elementary Teacher, College Teacher	2+	Informal	Online Search, Via Squeak	3
Ivy	F	Elementary Tech Coach	4	Informal	Online Search	1
Kent	M	Teacher Educator	23	Informal	MIT Media Lab	3
Larissa	F	K-12 Teacher, Teacher Educator	20	Formal	Online Search, Via Logo	2
Lenore	M	High School CS Teacher	13	Informal	Online Search	1
Clive	M	After-School Mentor	20+	Informal	MIT Media Lab, Via Logo	2
Jay	M	Teacher Educator	20+	Informal	MIT Media Lab	0.5
Sadie	F	High School CS Teacher	2+	Formal	Colleagues	0.5
Peter	M	After-School Mentor	2+	Formal	MIT Media Lab	2
Linda	F	Elementary Teacher	6+	Informal	MIT Media Lab	2+
Clayton	M	Elementary Teacher	20+	Informal	Online Search, Via Logo	1
Beau	M	High School CS Teacher	7+	Formal	Online Search	1+
Valerie	F	High School CS Teacher	5+	Formal	Online Search	4
Audrey	F	Elementary Teacher	3+	Informal	Colleagues	0.5
Jody	F	High School CS Teacher	8+	Formal	Educator Conference	1
Blake	M	After-School Mentor	1+	Formal	Online Search	1
Kristin	F	College Educator, After-School Mentor	30+	Formal	Online Search	1
Crawford	M	Middle School Teacher	6+	Formal	Online Search, Via Logo	4
Tara	F	After-School Mentor	1	Informal	Professional Development	1

Table 3.2 continued.

Name	Gender	Position	Years Teaching	Technical Background	Introduced to Scratch	Years w/ Scratch
Beverly	F	Teacher Educator	8+	Formal	MIT Media Lab	4
Georgia	F	High School Art Teacher	23	Informal	Colleagues	1
Arnold	M	High School CS Teacher	12+	Informal	Online Search	0.8
Kirby	F	High School Business Education Teacher	3	Informal	Professional Development	0.8
Jackie	F	Elementary Teacher	15+	Informal	Educator Conference	2
Taylor	M	Elementary Teacher	5	Formal	Online Search, Via Squeak	2
Candace	F	After-School Mentor	5+	Formal	Family Members	0.1
Sabine	F	Elementary Teacher	28	Informal	Online Search, Via WeDo	1+
Shauna	F	High School CS Teacher	1	Formal	Online Search	0.5

were based in the United States, with the exception of one educator from Asia and one educator from Mexico. All of the educators had experience working with Scratch in K-12 classroom environments, either directly or (in a small number of cases) indirectly as teacher educators.

Table 3.2 presents a demographic overview of the 30 teachers, including their interview alias, gender, position, number of years teaching, technical background (*formal*, such as a degree in a technical field, or *informal*, such as self-taught), how they were introduced to Scratch, and how long they have been working with Scratch.

As with the kids working at home in the Scratch online community, there are limitations and delimitations of the sample. First, the school context does not include the perspective of the students; the experiences of learners and their opportunities for agency are told from teachers' perspectives. This is intentional – in this work, experiences in both the out-of-school and in-school settings are told from the perspective of the people who are primarily responsible for the learning activities that take place. The practical reality of modern education and the majority of current classroom structures is that teachers control the space; hence, teachers were interviewed. That said, I think there is important future work involving conversations with young learners in classroom settings.

Second, Scratch is being used in a wide variety of settings – not just at home and in K-12 classrooms. For example, Scratch is being used for introductory computer science courses at the college level and in library drop-in programs. It would be interesting to compare the structures (and resulting learner agency) in these other settings. I intentionally focused on the home and K-12 classroom settings, however, as a way to focus on the assumed "extremes". Learning at home is often assumed to be low structure and high agency, while learning in classrooms is assumed to be high structure and low agency.

ANALYZING

Audio recordings from the interviews with kids and teachers were initially transcribed by graduate-student interns or by professional transcription services. I then reviewed the transcript drafts, checking against the original audio recordings. The transcriptions included every uttered word.

My initial plan for analyzing the interview (and related) data was to first thematize the data using the constructionist aspirations of designing, personalizing, sharing, and reflecting (described in the previous chapter), and then use the theorization of structure and agency (described in the opening chapter) to identify structures that enable or constrain agency. To what extent were kids having these types of constructionist experiences? To what extent were teachers designing learning environments that included these types of constructionist experiences? In each setting, what structures enabled their experiences? What structures constrained their experiences?

I performed an initial line-by-line analysis of the interview transcript data with NVivo 8, using this thematizing approach (Kvale, 1996; Miles & Huberman, 1994). I was able to code the data against the themes of designing, personalizing, sharing, and reflecting – but something was not resonating with me as I conducted the analysis.

I gradually realized that by coding against designing, personalizing, sharing, and reflecting, I was coding for *my* aspirations, not the aspirations of the kids and teachers that I was talking with, and that, in doing so, I was inadvertently undermining my theoretical emphasis on agency. My aspirations and desires, as manifested through the Scratch and ScratchEd activities, should only figure in the analysis as a form of structure, not as the organizing principle.

I discarded my analytical work and started again. Rather than imposing a framework centered on my aspirations on the data, I performed line-by-line coding of the interview data based on what kids and teachers described as important, their intentions and aspirations, as well as the tensions and challenges. I clustered the codes into common groups (different kids, different teachers used different language to describe shared experiences), and I then organized the codes by frequency to identify shared aspects of the experience (aspects described by all or many of the people I spoke with) or diverging aspects of the experience (aspects described by some or few of the people I spoke with). This coding/clustering/organizing process went through several refining iterations. The last iteration of interview codes, their frequency across sources, and how they contribute to the descriptions of each setting are available in Appendix B. This iteration formed the basis of the organization of Chapter 4 and Chapter 5, and guided the analysis of the other sources of data (observation and artifacts).

Chapter 4 – *Kids* – is focused on the experiences of young people at home with the Scratch online community – what they describe wanting, what gets in the way of their goals, what helps them to be successful in achieving their goals, and what validates their work. Chapter 5 – *Teachers* – is similarly organized, but focused on educators' experiences working with Scratch in K-12 classrooms. Both chapters provide rich descriptions of the intentions and tensions faced by kids and teachers, drawing extensively on the interview data, and supplemented by observation and artifact data.

Chapter 6 – *Agency/Structure* – revisits the description and analysis presented in Chapter 4 and Chapter 5, and draws out particular structures that enable or constrain the goals, aspirations, intentions, and desires of kids and teachers, in both out-of-school and in-school settings. These particular structures were identified through coding of kids' and teachers' descriptions of what they were doing, what they wanted to do, what they were unable to do – and the particular rules, roles, and resources that enabled or constrained them.

Chapter 4

KIDS

In this chapter, I describe how kids talked about their goals and aspirations for creating and connecting with Scratch – and the tensions that emerged when striving to achieve those goals.

Since the Scratch online community's launch in May 2007, hundreds of thousands of kids have found the online community and have started using Scratch without the support of structured learning environments, such as schools. Although for many kids their engagement does not extend past creating one or two projects, some kids have spent hundreds, or in some cases thousands, of hours at home developing and sharing Scratch projects. I wanted to learn more about these kids "in the wild", and over several years I conducted interviews with 30 active members of the Scratch online community.

A significant goal of the conversations with these kids was to understand what Scratch is from their perspective. How do they talk about what they do with Scratch? What aspects of their participation are important to them? What does Scratch mean to these kids?

The most significant theme across the conversations was the interconnectedness that the kids described between their experiences in the authoring environment and their experiences in the online community. For all of these kids, "Scratch" is equally about *creating* interactive media projects and *connecting* with others.

> It's a really great program that allows you to create your own games, animations, and be yourself and create your own media. So it's definitely a way you can be yourself, make whatever you want, and then, of course, there's an online community in which you can meet new people, be inspired to create your own stuff, and just experiment, and have fun.
>
> Clark, 12 years old

In this chapter, I describe how kids talked about their goals and aspirations for creating and connecting – and the challenges and tensions that emerged when striving to achieve those goals. The chapter is organized thematically into four sections, based on issues that kids highlighted as important and that recurred across the conversations. The first theme – *Enjoying Freedom* – highlights the freedom that the kids enjoyed both in product (the diversity of creation) and in process (the responsibility of defining and pursuing their own challenges). The second theme – *Getting Stuck* – focuses on the problems kids encountered during their open-ended creative design activities. The third theme – *Making Progress* – outlines the various strategies, both individual and social, that kids employed to overcome their creative obstacles and to develop as computational creators. The fourth and final theme – *Finding Audience* – shares the importance kids attribute to (and the difficulties they face in) seeking attention and finding audience for their creative work.

ENJOYING FREEDOM

17-year-old Lana has been using Scratch off and on for the past five years, her times of creative production coinciding with breaks from school. Her first project was a game, in which the player pops as many balloons as possible in a fixed amount of time. After this initial project, she created numerous other projects, exploring a wide variety of genres. She made interactive greeting cards celebrating holidays and friends' birthdays. She made animations with lush graphics and detailed plot lines, both on her own and in collaboration with others. She made tutorials that helped others learn how to create animated characters in her particular style. In all of her creative work, she follows her interests, lingering for as long as she wishes on a particular genre. Her most recent fascination has been composing music with Scratch, exploring and experimenting with Scratch's sound blocks.

11-year-old Jackson is newer to Scratch, having been a community member for little over a month. He loves exploring the Scratch website and trying out other people's projects, particularly games. Whenever he finds a game that he likes, he downloads the project and studies the code. He searches for places in the code that he thinks are not optimal. He tries to get rid of the noise around the "peaks" – the parts of the code he identifies as absolutely necessary for the project to run. Jackson takes each project as a challenge: can he reduce the number of blocks in the project without changing the appearance or experience of the game? He has done this "peak searching" with 10 projects so far, and he talks enthusiastically about the current (and future) challenges he is working on.

From Lana's original stories to Jackson's remixed games and everything in between, the collection of projects in the Scratch online community is highly diverse. This expanse of creative possibility is one of the first aspects described by all of the kids that I spoke with, in response to one of the initial questions in our conversations: "If you had to explain what Scratch is to one of your friends, how would you describe it?"

> It's really great to express yourself creatively. You could do anything with it. You can make video games, music, art, videos, anything. The possibilities are endless, no limitations, really.
>
> Lindsey, 12 years old

> It's a program that lets you explore your imagination. You can do whatever you want in it. You can create anything. There really is no limit to what you can make. You design your own stuff, and once you start you just don't want to stop because as you learn more, you can see there's more possibilities, and the more possibilities there are, the more you want to expand on what you just learned.
>
> Bradley, 12 years old

> Maybe that once you upload the whole working thing that you have a project. Or maybe it's just the creativity of Scratch.
>
> *What do you mean by that? Can you tell us a bit more about what it means to be creative with Scratch?*
>
> Well, it's just that there's endless possibilities. It's not like you can just make this project or this project and that's all that you can make.
>
> Nevin, 9 years old

> Well, I like that you can sort of do anything on it. It's like you can do whatever you want, really. You can be as creative as you want to be.
>
> Aaron, 10 years old

The sources of the kids' inspiration were as diverse as the projects themselves. Inspiration for projects emerged from personal interests – a passion for guinea pigs, an enthusiasm for soccer, a curiosity about linear equations or wanting to learn Japanese – and from the world around them. Kids talked extensively about the inspiration provided by their interactions with cultural artifacts – constructing a meme mashup based on two popular memes, Keyboard Cat and Nyan Cat, or reproducing popular mobile apps, like Fruit Ninja and Potty Racer, or creating animations extending the plot lines of popular book series, like Warrior Cats and Harry Potter – and from their knowledge of world events, such as the Tōhoku tsunami of March 2011.

They described the influence of others, within and beyond the Scratch community, such as accommodating requests for animated music videos of particular songs, making a game featuring squids for a squid-obsessed friend's birthday, or creating a game based on an older sibling's favorite video game. The Scratch authoring environment and online community figured prominently in kids' descriptions of inspiration. Particular sprites and sample projects often served as a creative basis for projects. Sometimes, however, the source of an idea was sudden, surprising, and unattributable.

> I can't really remember how the idea got started. Some of these ideas are just popping into my head. Like, I'm brushing my teeth and then all of a sudden, "Ooom! New idea."
>
> David, 9 years old

The freedom that the kids described enjoying was not limited to the diversity of creative products, but included the diversity of creative processes. As Feynman (1965) argued in an article about the learning of mathematics, "What is the best method to obtain the solution to a problem? The answer is, any way that works" (p. 10) – a sentiment shared by young Scratchers. The kids highlighted their experience of this freedom by comparing it with their other learning experiences, both in and out of school.

Scratch was not 14-year-old Lori's first programming experience, as she had taken an Information Technology course at her school the semester prior to discovering Scratch. In the course, the students were learning to use Visual Basic to create simple animations. Lori described the experience of coming to class and watching her teacher write out code on the chalkboard at the front of the room. There was one way, her teacher explained, of creating animations with Visual Basic, and he required that students use the code on the chalkboard exactly as written in their projects. When Lori asked him about other ways of making animations, her teacher discouraged her from worrying about that, stating that he had already provided her with the best way of making animations. Reflecting on her in-class experiences, Lori described feeling limited in her explorations of process compared with Scratch. "I've done a lot more in Scratch than I can in Visual Basic. With Scratch, there's like different ways you can make a project, but with Visual Basic, it's like, there's only one way that I saw that I could do things."

Other young Scratchers identified with Lori's experience. Jan described how he thought that the open-endedness of Scratch was fundamentally at odds with what he saw as the culture of school.

> I've met teachers who are very interested in using new technologies to learn and to do new stuff. Of course, it's a risk to try out new styles of learning or working because it always can go wrong – and that's why teachers are usually just a bit worried about changing, I think. Scratch allows students to actually do something themselves and learn to think creatively. They create as opposed to reproducing something they learned before. And the problem, of course, is that I don't know how valuable that is to most principals or teachers, but at least to me, that's I think the most important difference between learning with something like Scratch and learning at school most of the time.
>
> Jan, 16 years old

Out of school, the structure provided by knowledgeable others – often, parents with technical backgrounds and programming expertise – sometimes impinged on the open-endedness of Scratch. For example, Chuck was introduced to Scratch by his mother, who has been teaching computer science to college students for several decades. He spent hours on the family computer in the kitchen making projects. Occasionally, his mother would stop by and he would show her his Scratch discoveries and creations. He described the benefits he derived from being able to access her as a technical consultant – and the occasional tensions he experienced between his way of doing things and the way his mother thinks they should be done.

> Working with my mom – my mom is a teacher, so she "teachers" me things.
>
> *What does "teachering" look like with your mom?*
>
> It's basically – it's kind of subtle, actually. She just kind of, she'll basically watch me make something and then if she thinks there's a more efficient way, then she'll show me that. But, um – but secretly, I kind of like it better when she's not teaching because – I'm at a lot more liberty because she wants to do things the computer scientist way and I want to do stuff the – I don't know – the other, what would be the word for what I do? Just the way you do it easily in Scratch. I'm not really sure what the word for that would be.
>
> Chuck, 14 years old

Kids also talked about the open-endedness and freedom of Scratch in comparison to their other favorite activities. Jackson is – in his own words – "obsessed with MIT". He visits the campus whenever his parents are willing to take him and he follows the work of several research groups. Whenever research groups develop a tool that is publicly accessible, he downloads it immediately. He is a huge fan of a new game developed by a group at MIT, but described how he saw the game experience as different from his Scratch experience.

> [MIT-developed game] is fun, but you don't have much freedom – as opposed to Scratch where I have all this freedom. Like I can literally – I was just asked by a friend,

> “What is Scratch?” And I said, “It’s, um, it’s a program where you can do just about anything you want.” And that is a unique quality that I have never seen in any other video game or anything. There’s some rules, something to follow with other games.
>
> *Why do you like that freedom?*
>
> Because it’s more like real life.
>
> Jackson, 11 years old

GETTING STUCK

During my time as a doctoral student, I have often heard reference to “hard fun” – a phrase attributed to Seymour Papert, his way of describing the types of experiences that young people had with the Logo programming language. I was curious about the origins of this phrase and searched online for references. The phrase, clearly evocative, is referenced in thousands of articles, books, and other, more informal online missives. One of the top search results, however, is a short essay written by Papert (n.d.) about his introduction to the phrase.

> My whole career in education has been devoted to finding kinds of work that will harness the passion of the learner to the hard work needed to master difficult material and acquire habits of self-discipline. But it is not easy to find the right language to explain how I think I am different from the “touchy feely…make it fun make it easy” approaches to education.
>
> Way back in the mid-eighties a first grader gave me a nugget of language that helps. The Gardner Academy (an elementary school in an under-privileged neighborhood of San Jose, California) was one of the first schools to own enough computers for students to spend significant time with them every day. Their introduction, for all grades, was learning to program, in the computer language Logo, at an appropriate level. A teacher heard one child using these words to describe the computer work: “It’s fun. It’s hard. It’s Logo.” I have no

> doubt that this kid called the work fun because it was hard rather than in spite of being hard.
>
> Once I was alerted to the concept of "hard fun" I began listening for it and heard it over and over. It is expressed in many different ways, all of which all boil down to the conclusion that everyone likes hard challenging things to do. But they have to be the right things matched to the individual and to the culture of the times. These rapidly changing times challenge educators to find areas of work that are hard in the right way: they must connect with the kids and also with the areas of knowledge, skills and (don't let us forget) ethic adults will need for the future world.

Kids' desires for "challenging things to do" and the difficulty – and reward – of programming were recurrent themes in our conversations. Many kids talked about the great satisfaction and pride they felt in struggling toward and completing a project, like Lindsay, who talked about how she "learned how difficult programming could be, but how great the results are once you finally finish it." Others described fascination and engagement with the challenging process, like Jan, who felt that "if there's no challenge, it's not fun to create a project" and Allison, who described her Scratch programming experiences as "an engaging puzzle," saying that Scratch is "a lot of fun. It's challenging. It makes you think."

The exhilaration and frustration of creating Scratch projects came through most clearly as kids talked about their processes of developing particular projects. Extensive portions of our conversations were dedicated to discussing a project from their portfolio that they found particularly interesting. Kids were sometimes reluctant to talk about the challenges they experienced – either because they could not recall the details of a favorite project created months before or because of self-consciousness about not knowing. But I was always fascinated when kids opened up and articulated their struggles – and I found a conversation with one young Scratcher, 10-year-old Robin, particularly engrossing, as she frankly described the series of challenges she experienced while developing a recent project.

When I asked her if she wanted to tell me about a project, Robin described a project that she had developed in collaboration with her younger brother. The project was inspired by a game – 100 Levels – that she had played on a popular mini-games website. She recruited her brother to help her imagine the multiple levels of the game, and she quickly realized that it would be really difficult (or at least very time consuming) to come up with 100 different levels for the game. She and her brother decided that, at least initially, it might be more reasonable to start with 10 levels. Even with this compromise, she described how it "took us a really long time" to develop the project – "like off and on for maybe even a month" – because of the various challenges that she encountered in developing and testing each of the levels (e.g., there were problems with the laser beams, one of the levels couldn't be defeated). There were moments when she worried that she wouldn't be able to finish the project due to the limitations of her own experience and expertise with Scratch – "Sometimes I was just like, 'Oh my gosh, how do I do this? They aren't doing what I want them to, how do I do this?'" – and moments of challenge when she had disagreements with her brother. Robin eventually developed the project to the point where she felt comfortable sharing it online, but nevertheless felt frustrated. "It wasn't working as well at the beginning because I had a whole bunch of problems, and at the end there was still a problem because the backgrounds were sort of switched around."

For some kids, however, certain challenges were too great. The phrase "getting stuck" was mentioned by one of the kids in an early conversation, and it resonated with me; thereafter, I used this phrase when asking kids about their seemingly insurmountable challenges. Kids described getting stuck as the frustration of not knowing what to do or how to proceed. Edgar, for example, had an aspiration for a side-scrolling, platformer game, but expressed a deep annoyance with not knowing what the next step towards his goal might be.

> One thing that I don't know is scrolling. I have no idea how to scroll.

> *How do you think you're going to go about figuring that out?*
>
> I don't know. I've tried to make platformer games that don't use scrolling. But I have no idea how to scroll. I want to try to learn scrolling.
>
> Edgar, 12 years old

Adele also talked about the limitations of her expertise as a reason for getting stuck. An avid soccer player, she wanted to create a Scratch project that would simulate her weekly soccer drills. She hoped that the simulation could help her coach develop new plays for the team to practice. She imagined a project with a field, a goal, and a team of players – all of which she understood how to create with Scratch. She also imagined needing an "infinite" number of cones to place on the field for the virtual team of players to navigate around, but she wasn't sure what it might mean to have an "infinite" number of objects in a Scratch project and felt she couldn't make progress without understanding this part of the project.

All of the kids that I spoke with had been completely stuck on at least one project – the projects they "abandoned", "just left", felt "were too difficult to complete", or "gave up on". But the decision to abandon the project was easier for some than others and was motivated by different factors – from reluctance about further time investment, to negative feelings, to loss of interest.

> There's quite a lot of projects that I don't ever manage to finish.
>
> *OK. Why are those ones unfinished?*
>
> Well, either because I don't actually know how to. Or, I lose interest because I've thought of another idea. Or, often, I realize that I can do it, but it's going to involve a lot of complicated scripting.
>
> Fletcher, 14 years old

I had not fully appreciated the magnitude of abandoned projects until a site visit. Most of the interviews were conducted remotely via phone and Skype, so it was a rare opportunity to sit side-by-side with a young Scratcher and discuss their development processes and observe their (digital) work environment. I met 12-year-old Connor at a local school's Scratch open house event, and in a brief moment of quiet, I had the opportunity to be led through his collection of projects. He had shared a few projects on the Scratch online community, but had additional projects on his computer hard disk. I noticed two folders on his desktop: *Completed* and *Later*. The *Completed* folder contained 30 or so projects; the *Later* folder contained a few hundred projects. I asked Connor about the two folders.

> *What are all of the projects in the "Later" folder?*
>
> Projects I abandoned because, you know, I didn't know what to do.

Betraying more emotion than I intended, I asked him about the abandoned projects and he tried to console me.

> *But what if you really like your project idea?*
>
> It's just sort of sad that way.

MAKING PROGRESS

In an August 2012 New York Times op-ed piece, former software engineer Ellen Ullman mocked the Securities and Exchange Commission's suggestion to "fully test" stock trading software in an effort to prevent automated-trading errors. The suggestion, she argued, neglected a fundamental truth about software.

> There is no such thing as a body of code without bugs. You can test assiduously: first the programmers test, then the quality-assurance engineers; finally you run the old and new systems in parallel to monitor results. But no matter. There is always one more bug. Society may want to put its trust in computers, but it should know the facts: a bug, fix it. Another bug, fix it. The "fix" itself may introduce a new bug. And so on.

This stance on the presence of errors in software was shared by the young Scratch creators that I spoke with, and was often communicated in a similar no-nonsense, matter-of-fact way.

> There is no game that you can release with no bugs.
>
> Bradley, 12 years old

In the previous section, I described how the kids that I spoke with "got stuck" on bugs, and ended up *flailing* (not knowing how to proceed with a particular challenge and haphazardly searching for solutions) or *fleeing* (giving up on a particular challenge) – common occurrences when working with any sufficiently open system. These challenges require kids to develop strategies for making progress.

In this section, I describe the sophisticated strategies and dispositions that kids described as part of their project-creation repertoires, moving them from a stance of flailing or fleeing to a stance of *fixing*.

> Many children are held back in their learning because they have a model of learning in which you have either "got it" or "got it wrong." But when you learn to program a computer you almost never get it right the first time. Learning to be a master programmer is learning to become highly skilled at isolating and correcting "bugs," the parts that keep the program from working. The question to ask about the program is not whether it is right or wrong, but if it is fixable. (Papert, 1980, p. 23)

I share the strategies most frequently described by the kids, organized into two groups: individual strategies and social strategies. All of the kids talked about the tension between what you should be able to do on your own and what you should seek help on from others, a delicate interplay between relying on oneself and drawing on others.

> The creative process – it has really been evolving in me and learning from others. So, on one side, just learning from

> mistakes and investing time and changing projects, tweaking projects, and, on the other hand, help from the community.
>
> Jan, 16 years old

The individual strategies include: *experimenting*, *planning*, *compromising*, *persevering*, and *taking a break*. The social strategies include: *asking for help*, *studying projects*, *remixing work*, *working with others*, and *helping others*.

Individual Strategies

Experimenting

All of the kids that I spoke with talked about how important experimenting and exploring were as strategies for "getting better" with Scratch. Connected to their desire to have freedom in the development process, the kids described being curious and eager about figuring things out on their own by trying things out – from learning about elements of the Scratch interface, to understanding the effects of block stacks that they composed, to interacting with their Scratch creations.

These explorations were sometimes serendipitous. 10-year-old Evan described his process of creating a book-report project about *The Lion, the Witch, and the Wardrobe* by C.S. Lewis. He wanted to add an interactive component to his project – when someone clicked on the wardrobe, it should open – but he wasn't sure how to achieve that effect with Scratch. He knew it was possible, but he wasn't familiar with the mechanism. He talked about how his breakthrough realization came from exploring some other part of Scratch and then noticing a tab labeled *Costumes*.

> *Did you get stuck anytime while you were making this project?*
>
> I got stuck a few times. The one thing, it was really hard to – to make the wardrobe sort of look like it's opening. And it took a while for me to figure out that there were stuff called costumes. And I got stuck for thirty minutes on that. And, yeah, that's it.

How did you figure out that you could use costumes?

I looked at the Scratch website, and people used a bunch of sounds. So I found sounds up at the top – and while looking at the sounds and scripts, I found costumes.

Evan, 10 years old

The explorations were sometimes systematic. 11-year-old Jackson created a game in which the player navigates a cat through a series of increasingly challenging obstacles using the arrow keys on the keyboard. He described his biggest challenge – being able to move his cat up, left, and right, but not down – and how he overcame it by systematically exploring each block in the *Motion* category.

This got me so stumped. I eventually learned how to move a character with the arrow keys on this project.

What did you get stumped on? Figuring out how to move the cat?

Up and down arrow keys are – were – my biggest enemy. What I was doing was, instead of having *change y by* twenty-five, and *change y by* minus twenty-five, I had *move by* this and *move by* minus twenty-five. So, what was happening was I was going right, right, right, up, up, up, here, here, down, down, down. And it was basically not correct, as you can see. It's actually pretty efficient if you want to use diagonals, but I could not get my head around that I needed to *change y by* twenty-five. I could not get my head around that. So, I kept thinking that all I had to do is make him move in steps, but instead he had to move up and down the y-axis.

How did you figure that out?

I got unstuck because I went into this, *motion* [*navigates to the motion palette*], and I looked at this [*points to the list of blocks*] and I took out every single one of these [*points to the individual blocks*]. And I attached each and every one of them with my *when down arrow is pressed* block. And finally

> the character went down. And that was the one that I used forever.
>
> Jackson, 11 years old

Jenson, who often shared analogies during our conversation, talked more generally about his process of learning and debugging. Thinking across his project experiences, he compared it to Bananagrams, a popular tile-based word-construction game, which involves being flexible and experimenting with multiple configurations of letters to achieve the desired winning end-state. When I asked how he dealt with challenges, he described this flexibility and experimenting as his most reliable strategy when creating projects with Scratch.

> I usually just try a different way. Have you ever played the game Bananagrams?
>
> *Yes. Once or twice.*[1]
>
> It's like that – where you get so far into the program, or into the game, and then you realize, "Crap! I've got these other letters, how do I get them in?" And so you have to be able to take apart parts of your project and remix it up and put it back in and so that's just a way to think about it.
>
> Jenson, 11 years old

Planning

I have had the opportunity to talk with Lana several times over the past five years. In our most recent conversation, I asked Lana to tell me about a recent experience when she got stuck working on a project. She laughed, "I've actually had a few situations like that in my project when I just couldn't go further. I think I've learned the most effective thing to do is to prevent it." I asked her to tell me more about this evolution in her approach.

1 A fellow graduate student had recently introduced me to Bananagrams, and she and I had played the game obsessively during the weekend before this interview.

> My programming style on Scratch used to be, "It works. So don't complain." I'm used to just working to read the code in my project because it will be so individualized. Only I can understand it. It was really messy.
>
> *But you feel that's changed over time?*
>
> Yeah, that's changed. It's kind of like – I became more disciplined when it comes to coding. It's kind of a thing that comes with time, I think – with experience. Especially when you're working on complex projects, you have to keep everything organized.
>
> And you have to predict possible problems that can arise and have solutions ready. For example, when you're starting a music project, you should probably create a master volume variable and use that variable in all of the notes scripts when you need to set the volume for a certain note to be played. So that later on, when you need to silence the project for whatever reason, you don't have to go through all the scripts and attach the variable to all of the values.
>
> Lana, 17 years old

Kids described how they adopted more planning-oriented and systematic approaches to development as the complexity of their work increased, as a way of thwarting getting stuck. For some, planning involved diagrams and pseudo-code on whiteboards or sketching project requirements on paper, like the plan for a brick-breaker game shown in Figure 4.1, developed by a 6- and 9-year-old development team.

For others, overall development relied on a more bottom-up approach, involving experimenting and tinkering. But, even with this style, some still employed planning and organizing strategies – the code and scripts were broken up, organized, and annotated to keep track of the emerging modules of development, like 14-year-old Ashleigh who used comments extensively as she iteratively built up her project, to remind herself of the varying responsibilities of different stacks of code.

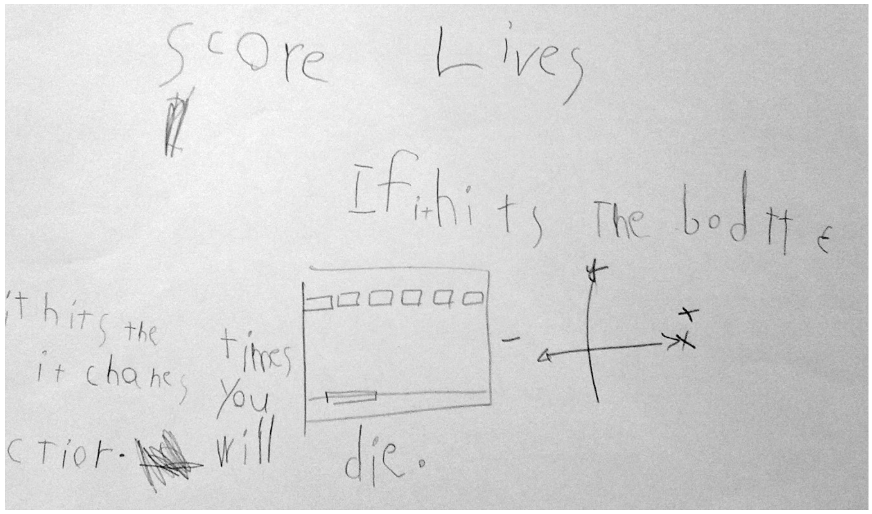


Figure 4.1
Project planning document developed by a 6- and 9-year-old development team.

Compromising

All design activities function within a set of constraints – constraints that can take on a variety of forms. I often think of design as a series of continuous approximations or compromises that work with/in these constraints. Similarly in Scratch, young designers are constrained and provoked to compromise, reframe, approximate. In our conversations, kids described how they worked with various constraints to rethink their creative visions – sometimes triumphantly, sometimes satisfyingly, and sometimes frustratedly.

Constraints were often imposed by the Scratcher's current abilities and fluency with Scratch. For example, the younger sibling of one of the kids that I spoke with talked about her desire to make a version of the popular *Angry Birds* game, which involves the player helping birds launch themselves at pigs who have stolen the birds' eggs. The sibling, who had only recently been introduced to Scratch, struggled with making a version in Scratch, given her lack of familiarity with the tool. In her version of the game, a bird could be launched, but would get stuck, suspended mid-air, and never reach its target. The birds, she decided, didn't seem particularly angry if they couldn't attack, so she named her project *Birds*, instead of *Angry Birds*.

For some kids, they possessed the technical capabilities and understandings, but the scale of the project was greater than the time they were willing to allocate. 10-year-old Aaron described his process of developing a music video for one of his favorite songs, and how he was flexible with his initial vision, in the interest of time.

> Well, the idea sort of changed because some things with the timing – some things just wouldn't fit in the timing. So, I had to sort of like take them out or add new stuff.
>
> *What do you mean, "It didn't fit with the timing"? Do you have an example?*
>
> Um, it didn't work with the timing parts. It kept taking me a long time to get the timing and I had to listen to all of the song just to get there, so at that point I just got a little lazy with the timing. That's why you see the sheep in a room instead of in a field.
>
> Aaron, 10 years old

Occasionally, the constraint was the capabilities of the authoring environment. For example, 14-year-old Matt talked about the limitations he experienced with Scratch's graphics, which currently do not scale elegantly.

> I don't know how small the stickman figure is, but I think I shrank him down to twenty-five percent and that's how I was actually able to get it to look like it was something. It was definitely shrunk down because of the – yeah, he shrank down to fifty-four. His head is flat on one side. You can see that. It's just – his head is flat on one side.
>
> *What's up with that? Why does he have a flat head on one side?*
>
> When you shrink the size like that, all the time, every single time it will deform the stick man.
>
> *I wouldn't have noticed had you not pointed it out. Does that detail bother you?*
>
> Not really. It's just something that Scratch does, so I just leave it. Sometimes it does bother me if I'm working on graphics. But that one wasn't about graphics.
>
> Matt, 14 years old

Persevering

Earlier in this chapter, in *Getting Stuck*, I described how kids had projects that they "abandoned", "just left", felt "were too difficult to complete", or "gave up on". Despite the challenges they experienced, the kids were incredibly resilient, bouncing back from one challenge to take on another, and talking more generally about the "persistence", "patience", "grit", and "determination" that they drew upon throughout their development processes.

When I asked kids what their advice for other kids who are new to Scratch might be, persevering with a challenge was the most common advice.

> Just keep going and make sure that you don't give up on your projects. Just keep rebuilding and constructing it and I think, pretty much, if you go on and on, it will pretty much work out.
>
> Jenson, 11 years old

> Definitely scripting, scripting, scripting. I was constantly experimenting, "Oh, this won't work," fixing it, experimenting, fixing it, and finally getting it right. Then having to make another version and fix more stuff. Patience is a virtue.
>
> Lindsey, 12 years old

The motivation to persevere through challenge was always connected to personal interest and passion. If someone is working on a project they feel a deep personal investment in, they will be willing to invest the time and effort needed. For example, 13-year-old Eva (the first to use the word "persistent" in my conversations with young Scratchers) articulated this as she described how much time a particularly beloved project was taking to complete.

> It took maybe a month or maybe a week to finish. I was working on it really persistently. Every day I would work on it. I would go to the computer and start working on it.

Oh, I love the word "persistently". Why do you think you were so persistent?

I really wanted this one to be done because I thought it would be pretty cool when I finished it. I thought it would be pretty cool and it is now. Actually, I had about 50 projects I was working on that were half done and had not finished any of them. I just thought, "I am not going to start anything new until I have finished something that I have started already" and that actually worked. I was like, "Oh my gosh, I really want to make this, so I really have to finish this first."

Eva, 13 years old

Or by 12-year-old Bradley, who described the great personal satisfaction and confidence that can come from prolonged effort.

You have to put work into it and it shows that you really do care about something. It shows that you were taking your time, you wanted to do something, and then it gives you that sense of accomplishment when you finally finish it. It boosts your confidence when you finish something that you thought was never possible.

Bradley, 12 years old

Taking a break

As important as persevering, however, is knowing when to take a break. Sometimes the breaks are brief, a way to clear one's head.

I usually save the project I'm stuck on and come back to it later.

I like that strategy.

Yes – take a break.

Brent, 8 years old

> When I was figuring out what functions to do, I'm like "This is a little confusing, let me just have lunch or something and I'll come back to it a little later."
>
> Aaron, 10 years old

> Sometimes, I get myself so dug in a hole that I have to just get myself out of Scratch for the day and relax myself and then start it up another time because after a lot of programming, your head hurts. It's not easy to program with a hurting head.
>
> Jenson, 11 years old

Sometimes the breaks are longer, making time to work on different projects entirely and build greater capacity with Scratch.

> A lot of the time when I get writer's block/artist's block, I take a break, do something else, and when I come back to it I am all refreshed and "Hey, an idea just popped into my mind!"
>
> *Is that usually what you do?*
>
> That's usually what I do. I usually just take a break, work on something else. I've even had cases where I take a break and work on a whole different project.
>
> Lindsey, 12 years old

Our conversations about these longer breaks in particular helped me think more optimistically about all of the projects that I initially feared were left incomplete. As I increasingly heard about projects left behind, I thought first of all of the unfulfilled dreams and aspirations. Certainly this is part of the story, but not its entirety. Kids talked about how many of the partially-completed projects served as material for future work. For example, David described how unfinished projects became collections of "extra ideas" that he saved in a special folder.

> *What do you do with the extra ideas?*
>
> I kind of make a tiny part for them and I can kind of go back to them. You know what? I'm just going to make a name for it, my Scratch workbox.
>
> *Oh, I love that – workbox.*
>
> I just thought of that right now. Ideas can help you very much.
>
> David, 9 years old

And, sometimes, the original vision is fulfilled after an extended break.

> I remember one project that I didn't give up on. It's called Tanks. I sort of gave up on it and then came back to it. I actually started making it a little less than a year ago and then stopped completely. Then right now, my friend saw it, and I revisited it and finished it and I actually found another glitch where player one, if they were running out of health, their health would go back up. And then I figured the glitch out and fixed it and now it's all good.
>
> Easton, 10 years old

Social Strategies

Asking for help

At the beginning of every interview, per our ethics board requirements, I talk with the kids about who I am, my interest in talking with them, their rights to answer (or not) any questions, etc. I emphasize my conversational aspirations for the interview, explaining that – despite the fact my interview partner and I have many questions – they are welcome to ask us anything at any time. Before diving into the main set of questions, I ask, "So, before we get started, do you have any questions?" Most kids are still somewhat shy at this point, and say no – interjecting questions throughout our time together or sending questions via email or chat later on.

But 9-year-old Adele caught me by surprise at the beginning of our conversation.

> *So, before we get started, do you have any questions?*
>
> Um…I want to know how I can use the variation list. Varis. The vari–
>
> *Variables?*
>
> Yeah, the variables.

My experience with Adele was very representative of how I understood the kids' strategies for seeking additional support. All of the kids I spoke to had experiences of asking someone else for help during their project development processes. In home settings, which were the primary work place for the kids I spoke with, parents and other older relatives were consulted for support in breaking down a problem and thinking through the challenges logically, if not for specific Scratch expertise. Siblings and friends were consulted for creative support, generating ideas for projects, and occasionally technical support.

Most kids, though, talked about their general isolation and lack of access to others with Scratch experience, and described the importance of having access to other creators through the Scratch online community.

> I don't know how it feels to be a new Scratcher now, but I know that I got very constructive responses on my projects, a lot of help from other Scratchers – and this help has been very important for me. I think if that wasn't there, I might have not, you know, continued in Scratch and be where I am now. So I think it was really important for me and therefore I think it's important that we support this kind of giving and receiving help in Scratch.
>
> Jan, 16 years old

This help comes in different forms and at different stages in the project development process, and in one's development as a designer

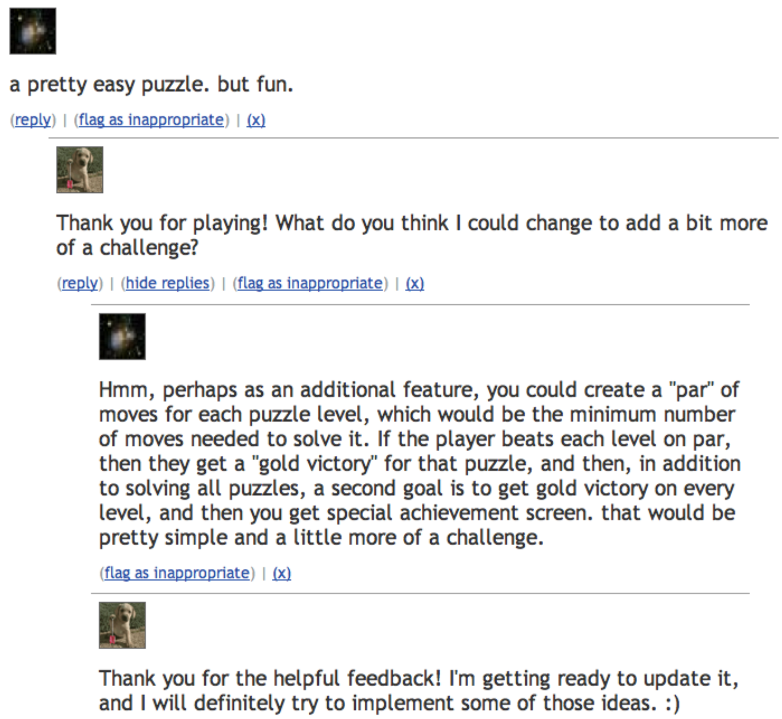


Figure 4.2
Use of project comments to solicit advice for improvements.

of interactive media more generally, which I describe in greater detail in a book chapter (Brennan & Resnick, 2012). For example, kids use the comments area beneath projects to solicit advice about how they could improve their projects (Figure 4.2).

The Scratch forums (http://scratch.mit.edu/forums/) are another important space in the online community where some kids are connecting with each other to ask for help about how to use particular features of Scratch, or how to deal with design challenges that they are facing. In the five years since the launch of the forums in May 2007, 23% of all registered Scratch members (approximately 275,000 of the 1.2 million) have used the forums, with approximately 10% (approximately 30,000 of the 275,000) of the forum participants having started or responded to more than 97,000 topics – for a total of approximately 1.3 million posts. There are many motivations for participating in the forums, however, not just seeking help with creative work, so I decided to focus on the interactions in one particular forum – the *Help with Scripts* forum – to take a quantitative look at how kids were giving and receiving support in the forums setting.

The forums have been restructured and reorganized several times since 2007, and although kids have long been seeking scripting support, the *Help with Scripts* forum was established formally in January 2012. In the seven months following, 863 people started 1449 topics, most of which correspond to someone asking a question. 4.8% of the topics were "unanswered", that is, had no posts in response. For the other 95% of topics, 668 people responded, with a median of 4 responses per topic. The median time to first response was 22 minutes, and the median lifetime of a conversation (from initial topic to final post) was 13.69 hours. In addition to people actively participating through responding, many others read the conversations, with a median of 126 views per topic.

Kids described two recurrent challenges that they experience in asking others for help. First, they sometimes feel unable to sufficiently articulate the problems that they are experiencing or what type of support they need. This is unsurprising (how do you talk about something you don't know how to talk about?), but nonetheless constitutes a barrier to successfully finding help. For example, consider the following (not atypical) exchange on the forums about how to construct a carnival game (Figure 4.3).

Second, even if able to successfully articulate their challenge and need, kids often find it difficult to make a connection to someone who can help them in a way that is accessible to them. Again, this is unsurprising. Many kids lack pedagogical experience and intuition about how to best support others, as we see in the continuation of the carnival game thread (Figure 4.4).

And, even with pedagogical knowledge, the content knowledge might be partial or incorrect. 10-year-old Aaron, for example, described the challenges he faced as he developed one of his games, and how it had required understanding absolute value, which he hadn't known about before learning about it from another young creator in the online community. He was particularly appreciative of having learned from this other Scratcher the multiple ways of calculating the absolute value of a number: (1) you could use the *abs*

#1

OK, I'm making a carnival game in scratch, and I;m having trouble at making some broadcasts only happen after other things have happened. My problem is, that I need to get my sprites to communicate with each other, basically, I want them to do this:

For shooter(s):
"when shooter (insert number here) clicked"
"broadcast (something)"

Then wall(s):
"When I recieve (something)"
"if I have been clicked"
"broadcast (something2)

For coin(s):
"when recieve (something2)"
"Glide to x:-240 y: -125"

This is what I want the sprites to do, I just dont know how to get them to do it because there will be 5 walls, 3 coins and 3 shooters and they all need their own coding, Please Help!!!

#2

I'm sorry but thats not very clear. If you could clarify what
you want to do, and maybe upload what you have so far, I may be able to help.

Figure 4.3 Typical forum exchange with someone looking for help.

#4

Use variables.

#5

how?
Can you show me or give me some examples?

#6

well if you make a variable for each sprites piece of data that needs to be communicated, and change them before you broadcast it would work fine.

#7

can you tell/show me in the forum pages as for some reason my computer wont let me download projects. Also, can you try and make it easier to understand as I dont...

Figure 4.4 Continuation of forum thread from Figure 4.3.

block in the *operators* palette, and (2) "you can just do zero minus the number."

Studying projects

In addition to connecting kids to other young creators, the Scratch online community connects kids to millions of Scratch projects. Every young person with whom I spoke had multiple stories of a project that inspired them and that they learned from. Studying others' code was a central practice for many kids, both as a way to get past particular challenges and as a way to learn more about Scratch in general.

12-year-old Bradley talked about how one of his initial interests in Scratch was to make space simulations, but he wasn't sure how to simulate the gravitational pull between planets. He posted his question on the Scratch forums, and someone advised him that he "would need to learn trig." He wasn't sure what that meant and searched for the term "trig" on Scratch, which led him to a list of projects tagged with "trig" (Figure 4.5). He downloaded several projects and studied their code, noticing that they all used similar *operators* blocks – *sin*, *cos*, *tan*. He experimented with those blocks and built small test projects, using them until he understood them well enough to produce the desired effects in his space simulations.

Several kids talked about how they used projects that were explicitly designed to support their learning – Scratch tutorials. Kids in the community have made thousands of Scratch tutorials, about all aspects of their Scratch participation, from the technical (e.g., how to use a particular block), to the aesthetic (e.g., how to create art in a particular style), to the social (e.g., how to become popular on the website).

But as with soliciting help from others, studying others' projects has challenges. Bradley, for example, although successful in studying others' code to learn about trigonometry, described how he felt confused by a tutorial that another Scratcher had created for him.

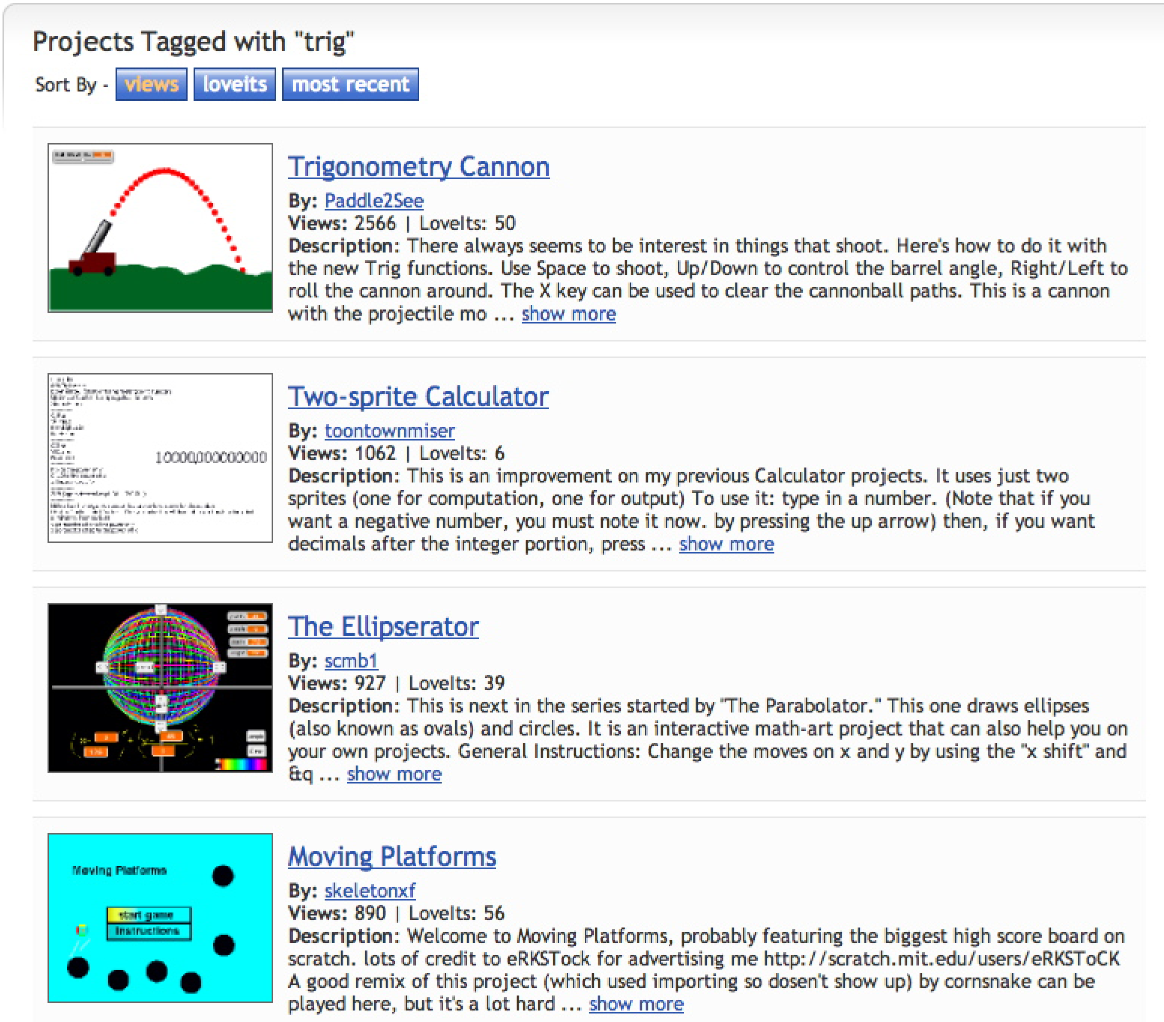


Figure 4.5
Search results for projects tagged with "trig".

> I wanted to make air hockey and it wasn't working out quite how I wanted it to. So, I had another user help me with an example project. But he made the example so complicated I couldn't understand it.
>
> Bradley, 12 years old

Others talked about the more general challenge of the legibility of unfamiliar code. When asked about reading other people's code, Ashleigh described these differences and difficulties in terms of one's "scripting style".

Why do you add comments to your code?

I thought for someone who doesn't know my scripting style, it would be easier to read if there were comments there.

I love the idea of scripting style, that's really nice.

When I download other Scratch projects, I can't read the scripts. They are too complicated for me. But my own are always like my own style – I know how to read them.

Oh, that's so interesting. How would you describe your own scripting style?

I don't even know really. I always use the same blocks and the same order for them and the same sort of – I broadcast at certain times and other times I don't. Other people are completely different in that way. When I see someone else's script, it makes no sense to me.

How do you think you developed a style?

It just kind of evolved, really. The very simple projects – like when you are first learning how to make them switch costumes and things – it's all the same for everyone. But when it keeps getting more complicated, I think everyone develops their own sort of way to go about it.

Do you think there are good or bad styles?

No, I mean, I think there are some that are different – but they are not necessarily worse. It's all just different.

Ashleigh, 14 years old

Remixing projects

Projects in the online community are intended not just for abstract inspiration or technical guidance – they are freely available to download and remix, to use the constituent components as building blocks. Just as professional programmers avoid "reinventing the wheel" by using libraries of code and common algorithms, young designers in the Scratch online community are free to remix and reuse Scratch projects to support their learning.

Almost all of the kids that I spoke with had a positive impression of remixing and appreciated having access to others' code – from the perspective of the learner or remixer.

> *How do you feel about remixing?*
>
> I think it's great. Because, quite frankly, that's how I learn most of my programming in Scratch – just by remixing a lot of projects. It's really great.
>
> Chuck, 14 years old

Kids talked about how remixing was an important entry point – and source of ongoing support – for their Scratch participation. It supports their technical development, allowing them to build from others' work rather than being intimidated by a blank canvas, or providing elements that they were uncertain how to program. For example, many kids are interested in developing side-scrolling games, but a basic side-scrolling implementation is fairly complex for someone who is new to Scratch. An experienced Scratch community member, recognizing the gap between interest and ability with side-scrolling, developed a project – both a tutorial and code foundation – for others to remix as the basis of their own side-scrolling projects (Figure 4.6). This project has been remixed hundreds of times (Figure 4.7), and has been folded into many different types of projects, including games, simulations, and art pieces.

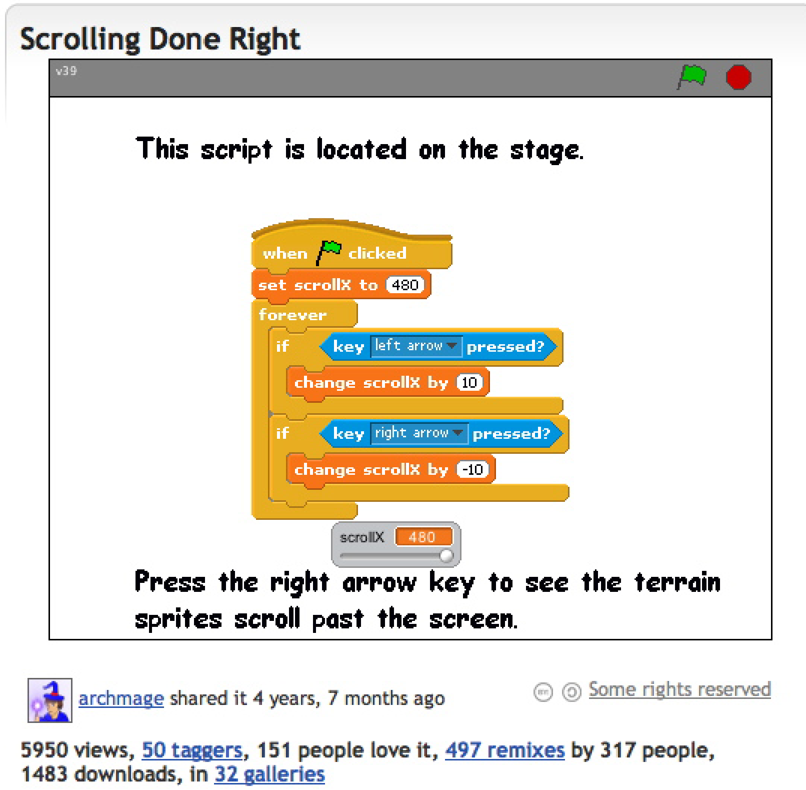


Figure 4.6
Side-scrolling tutorial project, intended for remixing.

Remixing also serves as an entry point for social participation in the online community, through pass-it-on or add-yourself projects, in which members are invited to contribute some small piece to an intentionally social project. This is a popular form of participation in the online community, with more than 10,000 projects having an explicit invitation to "Add yourself" – "Add yourself locked in the living room", "Add yourself falling from the sky", "Add yourself to the race", "Add yourself to the group photo", "Add yourself if you love Scratch", "Add yourself to the party", "Add yourself steppin' on the beach", etc. Through this, kids learn the basics of remixing as a community practice and of making connections with others.

Whether motivated by technical ends, social ends, or both, kids, although appreciating the benefits they derived from being the remixer, talked extensively about the frustrations of being the one remixed. It was seen as important for the remixer to both: (1) change the work sufficiently, and (2) acknowledge the creator through credit

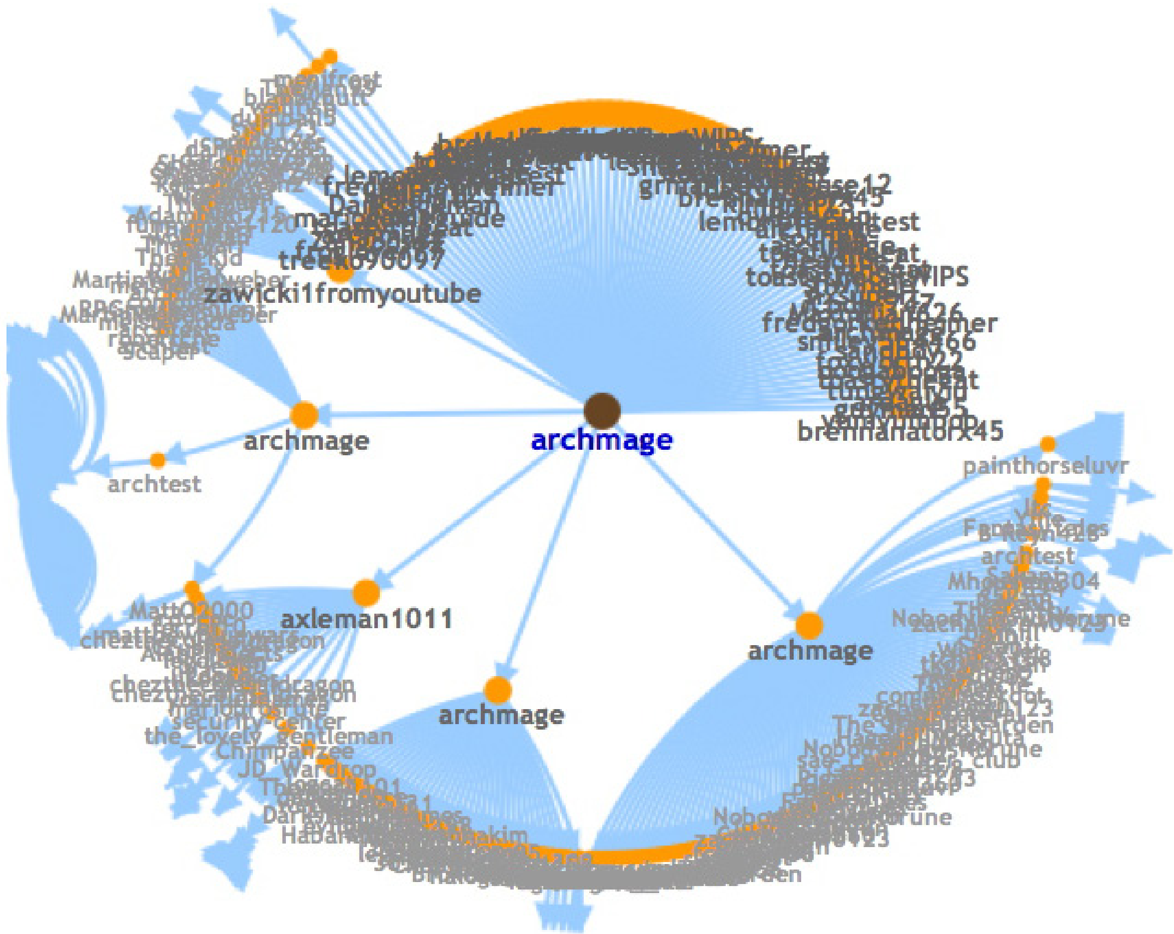


Figure 4.7
Visualization of the side-scrolling tutorial remixes.

and attribution. This attitude was succinctly summarized by 9-year-old Adele, when asked to explain remixing to someone unfamiliar with the concept.

> I would say that you take somebody else's project and you add your own little character or your own little spark to it. Then you put it back on and see how they like it.
>
> Adele, 9 years old

Minimally-derivative work was a particular frustration for kids who identified primarily as artists, because derivative works were heavily based on the artist's aesthetic signature. Eva, a Scratch artist who creates detailed animations based on the Warrior Cats series of books, was staunchly opposed to her work being remixed.

> *Have you ever had someone remix your projects?*
>
> Several times. And often without my permission.
>
> *How did that make you feel?*
>
> I usually yell at them a whole bunch until they get really mad at me and then I go away.
>
> *Why does it make you yell?*
>
> I think that they need to find something original. But they will be like "Oh, I don't know how to do this, so I will just take someone else's."
>
> Eva, 13 years old

Occasionally, derivative works *were* different, but were seen as problematically so, in that they took an idea and remixed it in a manner that the original author felt was offensive or deeply misaligned with the original vision. 9-year-old David, for example, was a great supporter of remixing in general, but was upset with how another Scratcher remixed his project about a friendly forest hedgehog. Instead of being supportive, the hedgehog terrorized his fellow forest-dwellers, and David griped about this change in theme: "He made my hedgehog evil!"

The importance of the degree of difference in the derivative work was eclipsed by the issue of credit and was a major source of kids' dissatisfaction with remixing. The expectation of appropriate credit is high (as documented in Monroy-Hernandez et al., 2011) – and the lack of credit was a constant complaint amongst the kids I interviewed.

> *How does it typically make you feel when someone remixes your work?*
>
> It makes me happy when they do that, but it doesn't make me happy when they claim everything for their own. When they

claim everything for their own, that's what makes me mad. Come on guys, you can give a little bit of credit.

Matt, 14 years old

Without sufficient credit (although there was considerable variation in what the kids identified as sufficient and appropriate credit), kids talked about these acts as "stealing", "copying", "plagiarism", and "theft". These acts are a significant source of strife in the online community more broadly, with numerous Scratchers flagging uncredited work and calling for greater control over remixing permission in future versions of the Scratch authoring environment and online community (Figure 4.8).

Figure 4.8
Suggestion from a Scratch community member to offer greater control over remixing privileges.

Working with others

Considering the benefit that kids describe in asking others for help, looking at others' work, and building on others' projects, a logical next step is combining these experiences, with Scratchers teaming up to create something greater than they could possibly create on their own. Very early in the life of the Scratch online community, and with almost no formal support or structure from the design itself, young Scratchers established creative partnerships (Brennan, Resnick, & Monroy-Hernandez, 2010; Brennan, Valverde, Prempeh, Roque, & Chung, 2011; Resnick et al., 2009; Roque, 2012). Kids who had experiences participating in group work discussed the technical benefits (e.g., learning new programming features from other Scratchers), but also emphasized the socio-emotional benefits – like 10-year-old Chelsey, who had been involved in one of the most prominent early collaborations (which she described as a "company") on the site.

> *What have been some of the greatest things about starting and working with your company?*
>
> Greatest things? I've met – I've made loads of new friends. It's really fun just making projects together rather than doing them on my own.
>
> Chelsey, 10 years old

The size of the partnerships varies from pairs to dozens of kids, and the structure of the group work is also quite variable. Some kids described approaching the work in a more cooperative, divide-and-conquer fashion, with different kids responsible for different elements of the creations. Many of the early, renowned partnerships followed this model, with certain kids serving in technical roles (the "programmers"), design roles (the "artists"), or in management roles (the "president" and "vice presidents"). Some of the cooperative endeavors were even more loosely structured, following a crowd-sourcing model. For example, 13-year-old Eva developed digital plays by writing scripts in Scratch projects and recruiting voice talent for various roles through Scratch-project-based auditions.

Others described adopting more collaborative, holistic strategies, with members of the partnerships taking on shared responsibility for all elements of the group production. 12-year-old Clark described his experiences of working with a group in this fashion, comparing his experiences to those of other high-visibility groups in the Scratch community.

> In our group, it's definitely shared responsibility. … I know there's [*Scratch-famous group*], and I've heard that they are one of the kinds that works separately on each project, and I kind of look down on that. Because, if you ask me, that's not even really a collaboration at its fullest. It's just, kind

of, giving everybody credit for just one thing, so everyone's really working like that. In our situation, we ask everybody to do something their own way, and then we'll compare to see which one is more efficient, or looks the best, like that.

Clark, 12 years old

The kids' enthusiasm for working with others was accompanied by a recognition that it necessitated a type of knowing beyond technical knowledge. As Jan reflected on his five years in the Scratch online community, he described the difference between creative and collaborative learning, and the real-world value of collaboration.

I think Scratch is something very good to do this creative expression and also learning programming. But it's not just learning programming. It's learning programming and being creative – and collaborating with different people online, which is like real life. You also collaborate in real life.

Jan, 16 years old

In our most recent conversation, Lana, who had worked with Chelsey in that early, prominent company, talked about essential strategies for – and challenges of – working with others on Scratch projects, none of which related directly to specific technical or creative abilities of the group members.

Basically, it just requires dedication on the part of the participants. Because it's not really a thing that people are doing out of goodwill – it's out of interest. It's always good when you have that enthusiasm because that's the only thing that keeps it going. And there are no ways to organize these things on Scratch – you can only communicate on Scratch via comments or forums. There aren't specific tools for organization and because of that, every person has to organize themselves. They have to be organized themselves. And that requires dedication.

You have to communicate. People in your collaborations should know exactly what to expect and what they want to

> achieve out of this. Because if their expectations aren't met and you're kind of forcing your own views on them and you aren't paying attention to their opinion, they'll probably get bored with the project. For that reason, it's always good to come to an agreement before you make a project. I think that's the main thing.
>
> Lana, 17 years old

Against her own standards for success, Lana considered the company to have been, to a certain extent, unsuccessful.

> I actually thought a lot about how to make organizations work, based on our company. I was thinking about why our company didn't work out.
>
> *Oh? Why do you think it didn't work out?*
>
> Well, there were a lot of reasons and mainly because, well, we were all inexperienced. Really inexperienced. And especially me because I was the person who was leading all of that. The whole thing. And I was in charge. And frankly in the end, it all boiled down to me taking over everything because I was so perfectionist about everything. I just think I wasn't communicative enough. That was, that was a really big problem.
>
> Also, I realized that the best way to work on projects is in stages. First of all, you develop the overall idea. Create a very rough sketch and you gather everybody's opinion about that. And when you're done with that, you can move on to the next stage which is basically, each time you kind of refine it, until you got a pretty good idea of exactly what you're going to do. And you start from the basic idea and then you go into the detail. And that way, you can ensure that nobody's opinion will be ignored. The problem was that a lot of people like to start from the detail, but they get confused because they don't know what to do. Then they start disagreeing and fighting with each other – and that's a really big mess. Because of that, you should always start from the big idea and after everyone

> has agreed on that, you move deeper into the project. And I think that's how projects and collaborations should be made.
>
> Lana, 17 years old

Helping others learn

Finally, whether creating tutorial projects or leaving constructive comments or answering questions in the forums, nearly all of the kids I spoke with had some experience with explicitly helping another young creator learn more about Scratch. I was curious about their motivations for supporting others – why did they take the time to help others?

For some, the motivations were deeply altruistic – they hoped that their work would be beneficial to the Scratch community. For example, 12-year-old Clark contributes actively to a community-generated wiki about Scratch, spending hundreds of hours composing and reviewing entries in the online Scratch compendium.

> *What motivates you to take on such a huge project – why do you do it?*
>
> I think it's because – it's like similar to the same idea why the creator of Wikipedia created Wikipedia, for a full source of human knowledge. And in my case, it's that, except about Scratch.
>
> Clark, 12 years old

Some kids viewed their efforts to help others as part of a process of generalized reciprocity, paying forward benefits that they had received (and continue to receive) from others in the community. 9-year-old Nevin, for example, was enthusiastic about how much she benefited from tutorials made by other Scratchers.

> There are a lot of great tutorials on the website.
>
> *What's a tutorial that a new person might want to look at?*
>
> Well, there's this tutorial that my friend made yesterday. I saw it on my "Friend's latest projects".

And what did it help you learn?

Well, it was just talking about the hat blocks and where you can import stuff, where you can bring in stuff.

Have you made a tutorial?

I did one on lists to add to the tutorials.

Nevin, 9 years old

Finally, kids talked about the benefits they derived from helping others. Helping others supported their own learning.

They say that teaching is the highest form of learning or understanding. So, I think that making math projects has actually helped me understand math concepts better than learning in school.

Sonia, 16 years old

Helping others resulted in positive emotional benefits.

It really makes me feel good inside to help people and see their projects and know, "Hey, I was a part of this, they may not have been able to make this, and this is awesome."

Lindsey, 12 years old

Helping others led to increased confidence in one's own development and creative practices.

Helping others gives you the sense that you are really – that someone's confused and you are helping them, and it makes you feel more confident when you try and do something.

Bradley, 12 years old

FINDING AUDIENCE

Most of the kids that I spoke with were very passionate about the importance of the online community – not surprising given that I was interviewing kids who were active participants on the site. But several kids were particularly adamant about the site's importance as a source of audience. Like Jenson, who, after I asked whether the site had been an important part of his Scratch experience, argued that sharing one's work was a necessary part of programming and of one's motivation for being a programmer.

> Every, I think, in my opinion, every programmer probably has a hope that somebody's going to view it.
>
> Jenson, 11 years old

This idea of creating *for* other people, as opposed to creating primarily for one's own gratification, was a recurrent theme among the kids. They were creating artifacts to engage other people – games to play, animations to watch, simulations to experiment with – and eagerly anticipated feedback from others, whether through views, comments, or love-its.

Receiving feedback was described as one form of being acknowledged for the significant effort invested in developing creative work.

> The approximate amount of time when you're making a movie I actually counted using my iPod stopwatch. It takes about 30 minutes to make. If you're making a good, well-detailed movie it takes about 30 minutes to make 30 seconds of screen time.
>
> *Wow.*
>
> So, it actually adds up to be quite a lot if you're making something. It's very hard work.
>
> *That's a lot of time.*
>
> Yeah. It needs to be appreciated.
>
> Jackson, 11 years old

But beyond praise and appreciation, attention from others in the community was valued as a way to improve as a Scratcher (as described earlier) – and served as motivation for ongoing work.

> What makes the online community good is most things – like being able to comment on other people's projects. Because getting feedback on something you made is a lot better than just making it and not having anyone see it. So, it kind of gives a purpose to making it. I don't think I would make as many projects if there was no feedback involved. It's just good to hear what other people have to say.
>
> Sonia, 16 years old

More Famous

Unfortunately, the hope for attaining attention and audience was often not matched by the kids' experiences – and the kids talked extensively in the interviews about the confusion, frustration, and sadness of not finding an audience in the online community.

> You know what I don't understand? Sometimes even though you do get a lot of views, that you don't get a lot of comments. And I don't really know why.
>
> *Why do you think that is?*
>
> I don't know, maybe it's just – like my mom said, it's like someone calling you and not leaving a message on the answering machine.
>
> Aaron, 10 years old

13-year-old Eva felt that she had enjoyed a reasonable level of visibility on the site – until she introduced Scratch to her friend, who very quickly attained a level of attention that Eva had not, which led to tensions between the friends.

> Yeah, I showed her Scratch. And she so rudely became more famous than me in like 3 days.
>
> *What is "famous"?*
>
> Getting a lot of comments or love-its, or whatever you want to call them. I usually check and see how many comments

> there are on a project – and other people are like "Oh my gosh, you've got like 1000 love-its. That's very famous."
>
> Eva, 13 years old

Eva's frustration and irritation were rooted in a sense of inequity – how could someone who had not dedicated as much time as she had achieve a greater level of visibility? Inequity was often the source of frustration for kids, whether it was someone who had spent less time, less energy, or had demonstrated less aesthetic sophistication, less technical sophistication, less attention to detail, yet somehow surpassed one's own level of visibility. 10-year-old Easton talked about this inequity as his least favorite part of the Scratch online community.

> You asked me what was my least favorite thing about the Scratch website. And now I remember something that I didn't before – how sometimes bad projects get on the front page while there are better projects that only have about five views. So, like, there were many projects on the front page that weren't anything as good as some of my projects, but they got front-paged. And also something that upsets me is that the Mario games always get front page even when they're really glitchy. Like, you can play the first level two times and you'll unlock level 3.
>
> Easton, 10 years old

Even kids who were not personally indignant about the lack of attention recognized it and suggested developing ways of distributing attention, like Sonia.

> Sometimes people get a little too obsessed with getting popular or famous, or whatever you want to call it, and get upset when they are not, and that kind of stuff. But there is a core group of people that are often on the front page and I think it would be good if that was spread out a little bit more.
>
> Sonia, 16 years old

And like Chuck, who discussed the consequences of a design change made to the front page of the Scratch online community, in terms of an individual's potential to be noticed.

> You guys [*the Scratch Team*] actually did this at some point: taking away the "New Projects" page. Because when that happened, I was really disappointed because that's a lot of people who haven't really gotten into Scratch yet. And it's how you actually get your work shown. Because unless you're like a really affluent person on the website, um, you're not – your projects really aren't going to get looked at. I really liked the "New Projects" part because it allows people who haven't really got a reputation yet to get a reputation.
>
> Chuck, 14 years old

Frustration with unattained or unequal attention also manifests frequently on the website through comments and projects. 10-year-old Aaron shared a project he created that highlighted his annoyance (Figure 4.9).

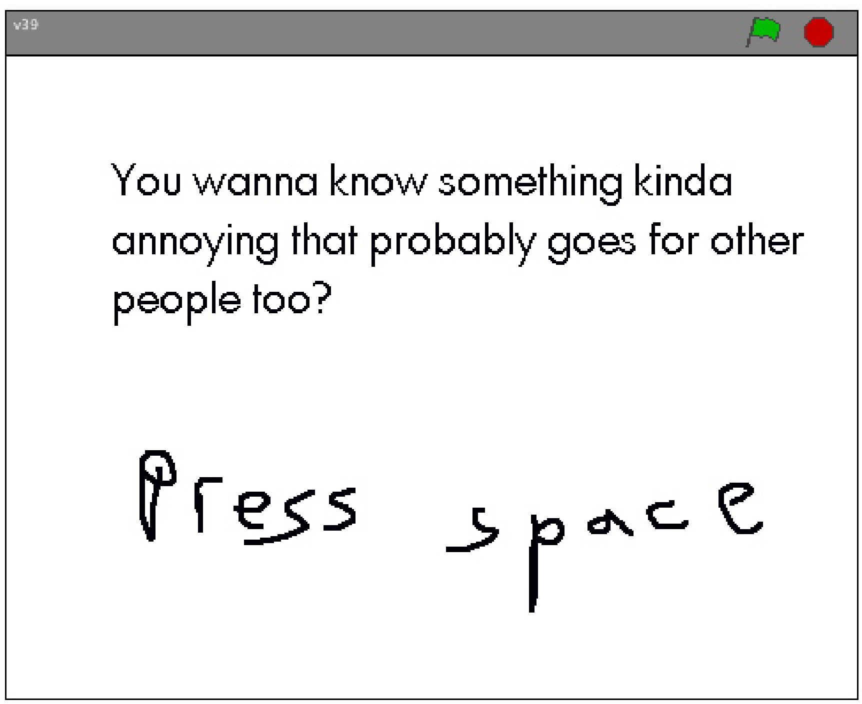


Figure 4.9
A project expressing frustration about attention inequities.

> You wanna know something kinda annoying that probably goes for other people too? You could make a great, awesome project but 99% of Scratchers don't even know it exists. But if [*famous Scratcher*] posted the exact same thing he would be on the front page in half a DAY. It annoys me because some Scratchers really do work hard and never get credit, and then [*another famous Scratcher*] makes a glitchy fall down and gets 1,848 comments and 599 love its. If you agree love it.
>
> Text from project by Aaron, 10 years old

The intensity and frequency of these comments in our conversations left me feeling both anxious and curious. The inequality of attention online is a well-documented phenomenon (Adamic & Huberman, 2002; Goldhaber, 1997; Lankshear & Knobel, 2002; Rheingold, 2012; Shirky, 2003). A small number of sources receive a disproportionate amount of attention, leading to power-law distributions (for example, Zipfian or Pareto distributions). Online attention is not the only phenomenon that follows this "80/20" behavior (20% of a population receiving 80% of the attention/wealth/etc., and vice versa). Newman (2005) demonstrates power law behavior in academic citations, words used in Melville's Moby Dick, bestselling book sales, and personal wealth.

To further investigate attention inequality in the Scratch online community, I developed a set of queries against the Scratch online community database. At the time I ran the queries, there were approximately 2.2 million visible projects. (I filtered out deleted or censored projects because it was no longer possible for them to receive attention.) Of these visible projects, 16% (348,466 projects) had only 0 or 1 views. But how were the 47 million total views distributed among the projects? Did most projects get the same number of views? Did some projects get more views? If so, how many more?

To visualize the in/equality, I use Lorenz curves. In a 1905 paper, Lorenz described how his approach to visualization illustrates inequity by plotting percent population against percent wealth. (In the context of the Scratch online community, attention is wealth.)

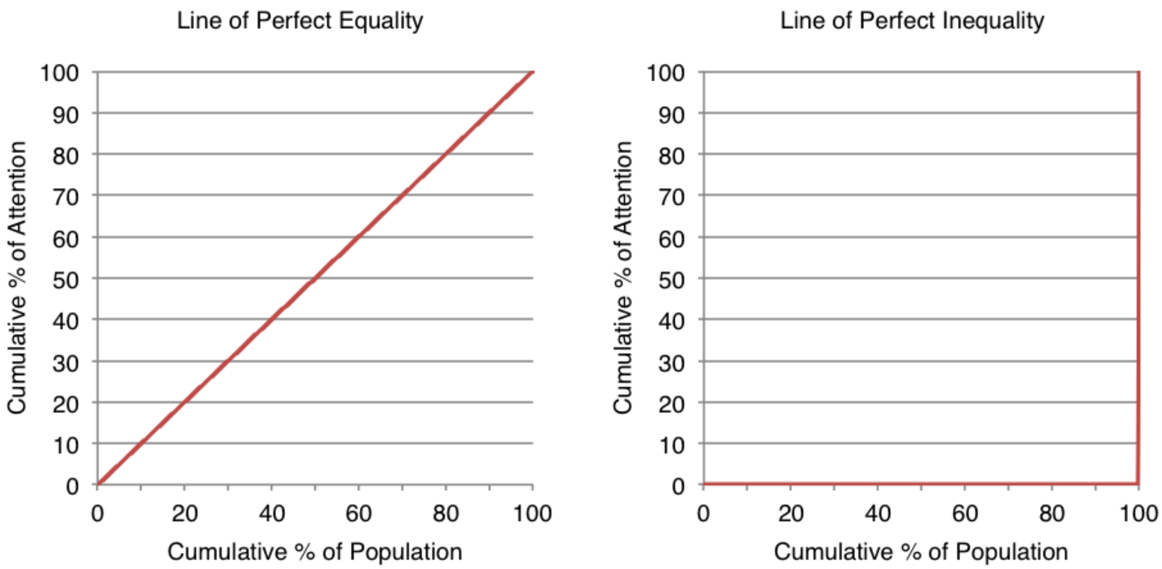


Figure 4.10
The lines of perfect equality and perfect inequality.

> The method is as follows: Plot along one axis cumulated per cents. of the population from poorest to richest, and along the other the per cent. of the total wealth held by these per cents. of the population. To illustrate, take a population in which wealth is distributed equally. … This will give a straight line. With an unequal distribution, the curves will always begin and end in the same points as with an equal distribution, but they will be bent in the middle; and the rule of interpretation will be, as the bow is bent, concentration increases. (p. 217)

Figure 4.10 illustrates the extreme cases – referred to as "the line of perfect equality", where N% of the population receives N% of the attention, and "the line of perfect inequality", where one member of the population receives 100% of the attention (Lorenz, 1905; Paglin, 1975).

For each project, I counted how many views the project had received, then sorted the projects from least viewed (0 views) to most viewed (94,990 views). Following the Lorenz method, I plotted the cumulative percentage of views against the cumulative percentage of projects. As Figure 4.11 illustrates, some projects *do* indeed receive more attention, with the top 20% of projects receiving 76% of the views. And the disparity was even more pronounced when I considered views by Scratch member (members who had posted at least

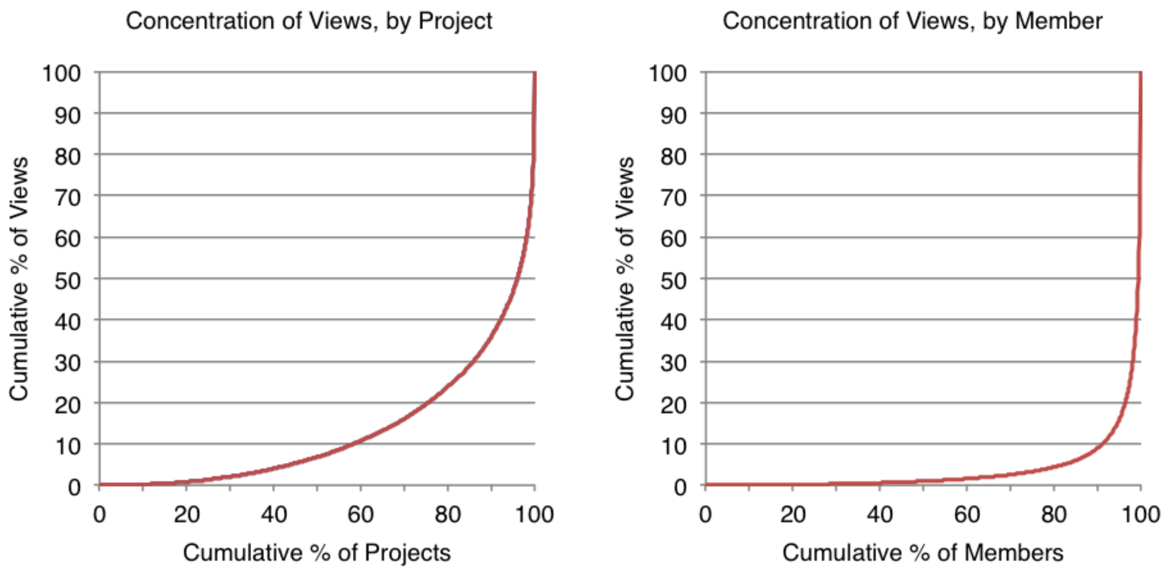


Figure 4.11 Concentration of views, by project and by member.

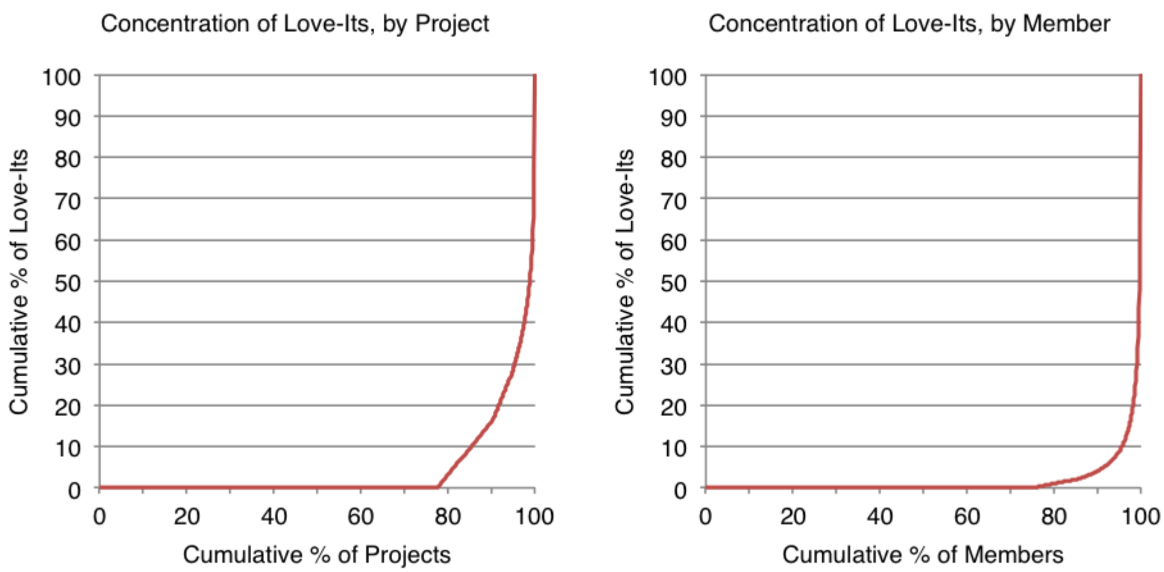


Figure 4.12 Concentration of love-its, by project and by member.

one project), instead of by project. The 20% most-viewed Scratch members enjoy 95.6% of the attention – which leaves 4.4% of total views to be shared among the other 80% of Scratch members.

Views are only one metric of attention, and in another set of queries, I examined the number of love-its that projects received. Love-its require an *intentional* click, and are a more robust signal of appreciation – kids can look at a project without loving it. Of the visible projects, 78% (1,717,830 projects) had 0 love-its. But how were the 1.7 million total love-its distributed among the loved projects and members?

Using the same approach used to describe views (above), I counted how many love-its each project had received, sorted the projects from least love-its (0 love-its) to most love-its (4086), and plotted the cumulative percentage of love-its against the cumulative percentage of projects. As Figure 4.12 illustrates, the top 20% of all projects received 97% of the love-its, while the top 20% of members received 99% of the love-its.

Getting What You Wish For, Sort Of

14-year-old Chuck was one of the first Scratchers that I can remember thinking of as "Scratch famous" – creating a project that catalyzed the interest and enthusiasm of the Scratch community. The project, which was a humorous, talk-show parody involving the Scratch Cat as a guest, was promoted as the first in a series, which in itself was an innovation in project-framing that Scratch community members were enthusiastic to support, both as model for their own creations and as encouragement for Chuck to produce many episodes. But he never produced another episode of the talk-show series. (He did, however, produce a project, three years later, explaining that he would not be creating another episode and requesting that people "please stop asking.") When asked about his experience with that level of attention, Chuck described how it was both beneficial and challenging.

> It's actually a funny story. At first, I was really excited about it because everyone really liked it. But what I eventually realized is that it would be really, really, really hard to do what I had promised, to actually be able to put another person on the talk show. And eventually, I basically said on the page, "OK, I don't think I can do this. Please stop asking me." But that didn't really work – no one really saw my comment. So, I'm still getting requests for people to be on the show even

though I still haven't made the second one. But I am really happy that so many people liked it. I think that's really cool.

Chuck, 14 years old

The project was shared more than five years ago and still continues to draw attention – and ire (Figure 4.13).

Look you got a lot of fans then you make an awesome project that everyone wants to be on then you just want to quit scratch by giving them a rip off you lost a lot of fans you just could make a project that says that your leving scratch well any way thats not how to get fans back from this rip off!!!!!!!!!!!!!!

Figure 4.13
Community backlash related to a discontinued project series.

A level of hostility often accompanies attaining visibility on the site, with featured projects receiving great praise and great criticism. As just one (but not isolated) example, a game-authoring Scratcher was frustrated by all of the attention that animation-authoring Scratchers were receiving in the community, as he felt that they were not sufficiently programming-focused. He created a protest project, encouraging people to boycott all animation projects and calling for a return to a programming focus (Figure 4.14).

But even for less "famous" Scratchers, negative attention was described as a problem, with comments from others not always being as respectful or kind as kids hoped. These problems are shared through projects and on the forums, and communicated to the Scratch Team via email.

Subject: Scratch is becoming a bad site :(

To: Help@Scratch MIT

I really like scratch, I've used the Program for many years. I was last year shown to the home site. It was a great place, I found many cool projects, people, and ideas. Through scratch, I have acheived great personal growth. But that was last year... even last year this was kind of evident, but the problem has grown. Its every where I go now, there Isn't a single time I log onto scratch and dont run into one of these problems... There are Haters on this site...lots of 'em. They go

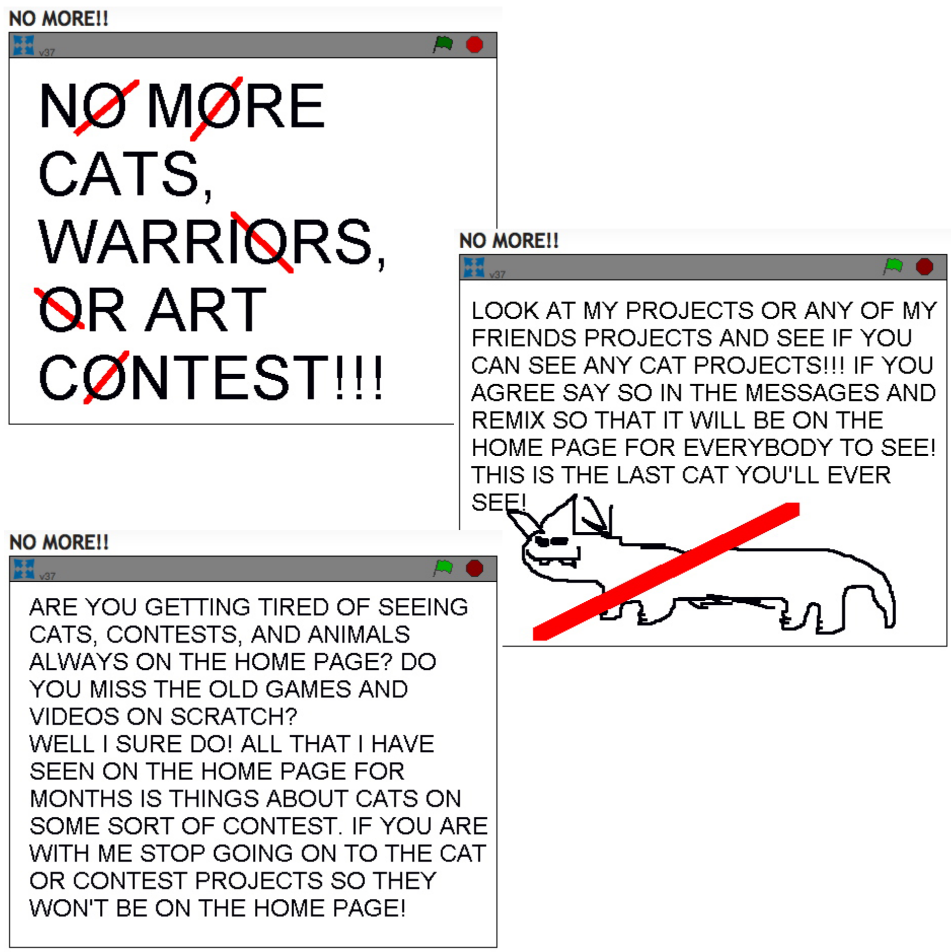


Figure 4.14
A Scratch project protesting the perceived domination of the front page by animation projects.

around down grading any one they dont like. I even got a hate message, and it was the first message I got on my first project. There is also lots of bad words on this site. Scratch is primarily a site filled with little kids. But there are a bunch of famous, or semi-famous scratchers out there that are older, and they use bad words. some of them over use bad words. Being that you have your eye on the site (I hope) I'm pretty sure you know who they are. Theres a lot of fighting, cussing, and stealing on the scratch website. That sounds more like a a description of a city filled with thievs and drunkards if u ask me. This site is filled with a bunch of kids and they are obviously acting unapropriatly. I know that your server is suppost to be a place where we come together to share create and learn to program {with fun} together. But scratch just

> isn't that place. And lots of people are quitting and getting thier fealings hurt because scratch just isnt that place. I know that you cannot change the people that use your server, but there must be something you can do to try to change all this disgusting behavior...right?

The challenges of online interactions are well documented. Online social interactions match the complexity of in-person social interactions, and can make interactions even more challenging, given the lack of signals and cues to support understanding an individual's meaning and intentions (Donath, 2007). This complexity can lead to problematic interactions between people online, including both unintentionally and intentionally problematic behavior, such as conflicts, bullying, trolling, and drama among community members (Golder & Donath, 2004; Kollock & Smith, 1996; Marwick & boyd, 2011).

I heard about some of these challenges in my conversations with the kids, and learned about how they dealt with negative interactions. The kids described how they relied on the affordances of the website infrastructure and on Scratch Team members to intervene, through flagging or ignoring features. But, most often, they described the ways in which support from other community members helped provide structure and guidance for appropriate behavior in challenging times. 9-year-old Adele described her experience of observing this community support in action.

> There are a lot of people that are willing to help you and it is a very safe website because I saw once something when somebody said like they didn't like the person's project. About 8 people said, "That's not nice." So that impressed me that so many people cared.
>
> Adele, 9 years old

8-year-old Brent told me about his own experience of how positivity from the community balanced negativity from an individual.

What's the first thing you do when you come to the Scratch website?

It's usually my projects. Because I just made a project and I want to check on my project. Because I've already gone and I saw a lot of negative – I feel like I've gotten bad comments.

Have you thought about disabling comments?

No, I don't really want to disable it because I just want them to be a little – I usually like the comments. That way they can say nice things about it.

[Looking at comments] I see – because there are lots of nice comments, too.

Let's take this comment right here. I like it because some people are standing up for me. So I usually don't delete a mean comment if somebody is already standing up for me.

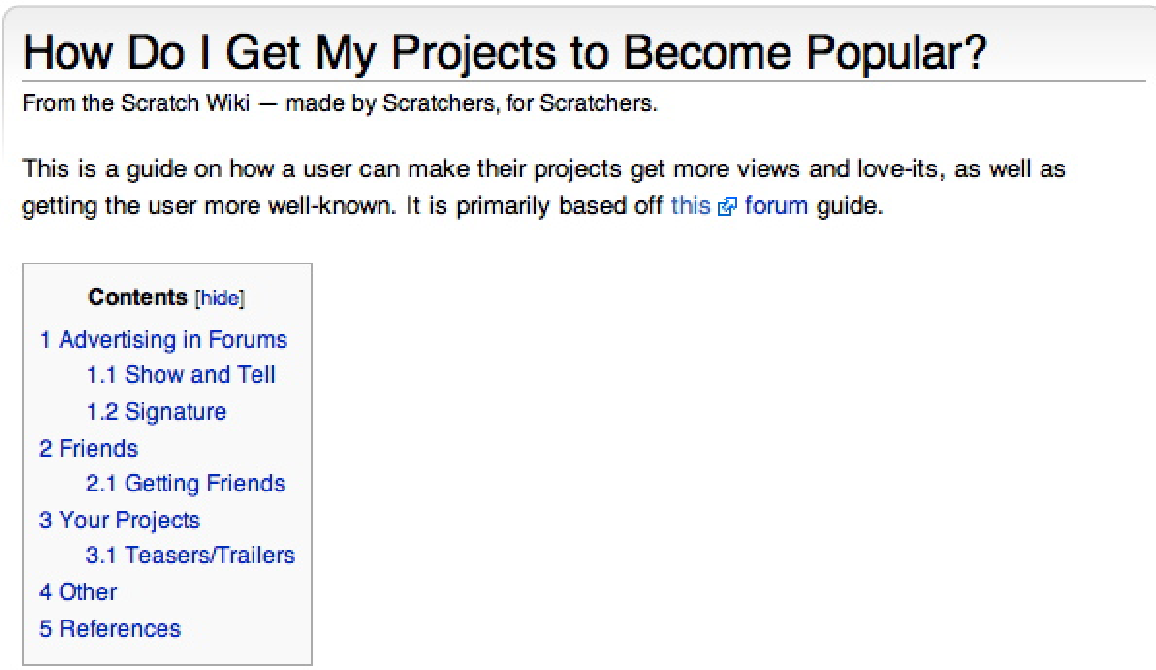
Oh, that is nice. You can always flag it and we get an email about it so that we can let that person know that it's not OK to say things like that.

I'm OK with it as long as – I don't really want them to comment like that again, but I'm OK with it if somebody is already sticking up for me.

Brent, 8 years old

Developing Strategies

Despite the challenges of finding audience and developing social interactions, the kids value it highly, and are thinking strategically about how one cultivates audience. This thinking is often communicated through projects and forums – *many* projects and forum posts. A search for "how to animate" on the Scratch website yielded 2,670 results. A search for "how to scroll" yielded 5,470 results. A search for "how to famous" yielded 4,460 results and "how to popular" yielded 5,310, just two of the words that kids use to describe visibility.


How Do I Get My Projects to Become Popular?

From the Scratch Wiki — made by Scratchers, for Scratchers.

This is a guide on how a user can make their projects get more views and love-its, as well as getting the user more well-known. It is primarily based off this forum guide.

Contents [hide]

1 Advertising in Forums
1.1 Show and Tell
1.2 Signature
2 Friends
2.1 Getting Friends
3 Your Projects
3.1 Teasers/Trailers
4 Other
5 References

Figure 4.15
Scratcher wiki entry on popularity.

Some of the recommendations are thoughtfully earnest, like a young Scratcher's project recommending a four-step popularity process ("Step 1: Making a decent project, Step 2: Adding lots of people to your list of friends, Step 3: Adding your project to a gallery, Step 4: Using a popular tag") and the Scratcher wiki entry on popularity (Figure 4.15).

Other advice is slightly more cynical – such as "Tips for Scratch Fame" (which advises the judicious use of spam to promote one's work, and advising to not "be sad if you don't get famous, because you probably won't"), "How to get famous" (which advises the construction of fake apology projects because "apologies are an easy way to get the front page"), and "Tips'N'Tricks" for how Scratch works (which advises luring people in with highly engaging, if not entirely accurate, project thumbnail previews) – as illustrated in Figure 4.16.

Several of the kids that I spoke with talked about the role of reciprocity in acquiring attention, both as a way of moderating one's expectations and as a way of shaping one's behaviors. To receive attention, one must bestow attention – and it is not reasonable to expect otherwise, as Matt explained.

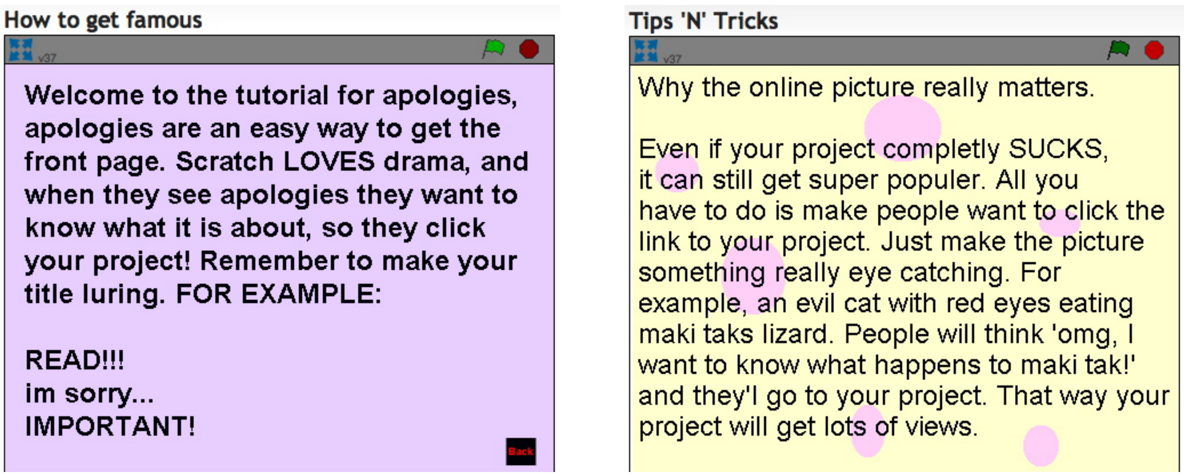


Figure 4.16
Cynical advice from Scratch community members for increasing visibility.

> I don't know what's wrong with people, but everybody keeps quitting for popularity. I just don't understand it. They're like, "Oh, I've been gone for a week and I had one message, or I had no messages." Well, that's because you didn't reply to anybody else. You didn't talk to them. If you don't talk to them, they're not going to talk to you. You don't just expect people to randomly e-mail or give you a message or something. You don't expect people to just go ahead and on one of your projects and start commenting on it and give you several messages, stuff like that. You don't expect that.
>
> Matt, 14 years old

Ashleigh described engaging in more generalized reciprocity, a behavior stemming from her disappointment in not receiving comments. A lack of comments on her projects served as motivation for reaching out to others to lessen their disappointment.

> *What makes you want to leave comments?*
>
> I see like a lot of views and not that many comments on my project and somehow, not even disappointing, but yeah, I guess disappointing for me, so I like to leave comments on other people's projects so it's less like that.
>
> Ashleigh, 14 years old

Sonia described her connections to others as an extremely important part of her experience and she wished others could have similar experiences. I asked her how others could develop those connections.

> *So, what advice would you give to a Scratcher who isn't maybe connected to people?*
>
> I don't know – it's hard. Commenting on other people's projects makes them want to comment on your projects, so that helps. And adding your projects to galleries or advertising them on the forums, that kind of stuff. I don't know. I feel really bad for people who never get views, so I try to look for ones that don't have many and comment on those. Sometimes you just get lucky.
>
> Sonia, 16 years old

Changes in creative and social practices, as shaped by a desire for attention, audience, and popularity, were not always aligned with the kids' other desires. For example, 13-year-old Eva created animations (based on the Warrior Cats book series) that were highly regarded in the online community. When I asked about her source of inspiration, I was surprised by the response.

> *I assume you like Warrior Cats.*
>
> I don't really read the Warrior Cats books. I read the first series, but they weren't actually that good in my opinion. … A lot of people on Scratch like to read the Warriors series and they are big fans. I thought this might be a good way to get viewed.
>
> Eva, 13 years old

Sebastian talked about the attention economy of Scratch – how, given the limited attention available, he was more strategic about when he shared his projects.

So, you don't tend to post early and then change it?

Yeah. I mean I was doing that and then I realized that's, I mean – I know this is not the point of Scratch, but I think it is really cool when you get a lot of views, because then you know people are seeing your projects, and they are enjoying your project. And so, when you post it half-done, people will check it out, and they will say there's a lot you can do here, and then you never see them again. So, it's sometimes like a pattern, like the projects that get on the front page, they're always like really, really finished. So, I try to do that. It's really hard at first. Like you just want to post it and get feedback, but you won't until you get it to a really good point.

Sebastian, 13 years old

Chapter 5

TEACHERS

In this chapter, I describe tensions that teachers identified between their aspirations and the actualities of implementing Scratch in the classroom.

In the previous chapter, I described the experiences of kids who work with Scratch primarily at home. While there are many kids who work with Scratch this way, an increasing number of kids are being introduced to Scratch in other learning environments, by teachers – and, in this chapter, I turn my attention to these teachers. I take a broad view of "teacher" here, referring to any adult who supports young learners, from K-12 classroom teachers, to after-school-program facilitators, to teacher educators.

I wanted to better understand the experiences of these teachers and what Scratch looks like in their settings, particularly in relation to the understandings I have been developing about kids working with Scratch outside of schools. How do teachers think about Scratch? What are their experiences of incorporating Scratch in the design of their learning environments? To explore these questions, over the past four years, I conducted interviews with 30 teachers who have been working with Scratch.

A central theme in our conversations was the challenge of incorporating open-ended design activities in these learning environments. Creating opportunities for students to understand and to build capacities for engaging with this type of open-endedness was described by many of the teachers as the central aspiration of learning and education, yet in conflict with how learning is enacted in many school settings.

In this chapter, I describe tensions that teachers repeatedly identified between their aspirations and the actualities of implementation. The chapter is organized thematically into four sections. The first theme – *Supporting Problem-Solvers* – unpacks teachers' motivations for working with Scratch. The second theme – *Negotiating Open-Endedness* – explores the challenges teachers face when trying to implement open-ended design within the structure of the classroom. The third theme – *Building Culture* – outlines strategies that teachers have developed in response to the challenges that accompany open-ended design activities. The fourth and final theme – *Legitimizing Learning* – describes the challenges teachers experience when trying to understand and explain the learning that is taking place in their learning environments.

SUPPORTING PROBLEM-SOLVERS

Michael Smith-Welch is a graduate of the research group to which I belong at the Media Lab. Several years ago, during a conversation that we were having about Scratch, Michael introduced me to a beautiful set of questions – originally with *art* as the subject of the questions, but easily translated to other topics, including Scratch. *What is Scratch? What is Scratch good for? What is good Scratch? How do different people think about these questions differently?* I have subsequently used these questions in a variety of settings, including my conversations with teachers about their Scratch experiences.

When asked what Scratch *is*, most teachers that I spoke with responded that Scratch is a programming language. When asked what Scratch is *good for*, most teachers described the value of learning how to program as a way for young people to connect with computational culture. Young people need to develop better understandings of and fluency with the technological landscape that surrounds them.

> It's the future for these kids – that's the way things are going to go. It's going to become more technology-based, and they already have a head start on it that I certainly didn't have. That's just common sense. It's the way the world's going, and they need to be up on it.
>
> Sabine, Elementary School Teacher

> I give the talk that, "You might probably end up not being a programmer – maybe some of you will, but most of you won't. But you really want to be able to have a really deep understanding of the power of programming, so that you say, "Maybe I can't do the programming for this research project, for this graphics project, but I know I can talk to the geeks or, you know, whoever. I know the power of what they can do."
>
> Clayton, Elementary School Teacher

Although Scratch is a programming language, the focus for many teachers was described neither as programming for programming's sake, nor as programming in the exclusive service of understanding the technological world. In a story on ScratchEd, an educator from Australia described her first experience teaching Scratch and her explanation of Scratch to parents.

> Parent/teacher interviews. Picture a hall, set up similar to "speed dating". But replace hopeful singles with concerned parents on one side of the desk, and well-intentioned teachers on the other. It's getting late, my 20th parent tonight sits down. Mr. Smith introduces himself as a parent of a student in my IT class, and asks the questions that many other parents have been asking this year. "Why Scratch? Why teach my child programming? How can you expect to teach programming to children?"
>
> My answer always refreshes my own belief in why I have come to love teaching Scratch.
>
> I say, "Mr. Smith, when I first saw Scratch I wondered how many of the students this unit of work would reach. I mean, even if one student becomes interested in computer programming after this unit, is that enough to teach programming to an entire class? From a programming point of view, Scratch takes all of the essential constructs like sequencing, conditional branching, control structures, data manipulations and places it in an easy environment, which every student is able to use as simply as using children's building blocks. What you need to understand is that Scratch teaches much more than computer programming. Scratch is important because it is about teaching students to solve their own problems and getting them to figure things out and discovering how to work things out for themselves."
>
> When asked about Scratch, students gave many suggestions as to its importance:
>
> "Scratch taught me to fix problems on my own."
>
> "It made me be efficient with my time."

> "It taught me that I need to try new things when it didn't work the first time."
>
> "I took pride in my work, because I had done it all by myself."

This theme of Scratch and programming as a context for supporting the development of problem-solving capacity and creative thinking was the most common thread across my conversations with teachers, despite the lack of uniformity in their teaching settings (working with learners of different ages, and with different disciplinary intentions). Programming is a rich context for developing creativity in problem-solving abilities –

> I really try to work the creativity side – how can you creatively solve a problem? Maybe problem solving is a better way to say it. Of the kids I work with in my AP Java and this class, probably between those three classes, there are 70 kids. Maybe 12 of those will be programmers or computer scientists. The others, I want them to have a love for the computer and a love for methods of solving problems. Because that's what life is about, really. I can't do everything, but I can pretty much figure out most things through problem solving. And that's what I am trying to get across to the kids.
>
> Arnold, High School Computer Science Teacher

> I was concerned about teaching a programming course to my kids, not only because I didn't know much about it, but because how much were they going to use programming? Were these kids even interested in programming? A lot of them aren't and they're not necessarily looking for a programming class. But then I got to looking at Scratch and noticed they can learn a lot more, even if they are not interested in programming itself. Problem-solving and troubleshooting skills – "Why isn't this working?" and "How can I get it to work?" The whole basis of the program is having that challenge before you, knowing that, "This has to work somehow. I can get this to work and this program to do what I want it to, just how do I? What do I need to do to do that?" And so,

> even if they weren't interested in programming, then that was a valuable skill to learn.
>
> Kirby, High School Business Education Teacher

Programming is also a rich context for developing creativity in expressing oneself through designing and making.

> For most of the kids, especially at the elementary level, this is the first time that they have had any experience with programming at all. So, I don't worry about the programming so much – it's more about the design process, and figuring out problem-solving strategies. They get to create and express themselves. The kids are creating something new that they enjoy and they feel is what they basically wanted to create. They should feel successful. If the kids end up with more confidence that they can take on a challenge – more confidence, whether it's with computers, or being able to tackle challenging problems or as they work through to make their creations, you should feel successful.
>
> Taylor, Elementary School Teacher

> What has surprised me the most, not having a background in technology or computer science, is that I wasn't expecting to enjoy programming and teaching it as much I do. It's so creative – and I don't think people realize. I don't think all my students, who aren't familiar with Computer Science, the ones who don't take my classes, I don't think they realize how creative it is. And so when my students come and they come to Computer Science at the beginning of the year, they don't expect that they're going to be making these crazy things! And they start to really get into it, and it just makes me happy to watch them be so motivated and I'm not doing anything to motivate them. It's opening the possibilities for them to create something that motivates them.
>
> Lenore, High School Computer Science Teacher

NEGOTIATING OPEN-ENDEDNESS

Teachers shared their aspirations for the types of learning environments they desire to create for their students – learning environments that engage their students in creative, problem-solving activities. How to create these environments is a source of considerable tension, as teachers reflect on their own practice and debate approaches with each other. Such a debate sprung up at a Scratch workshop that I co-facilitated several years ago.

The three-hour workshop, hosted at a regional technology conference for teachers, was framed as an introduction to Scratch. After the 20 participants arrived, we showed them three or four projects created by kids, to give them a sense of what might be possible to create with Scratch. Then, we transitioned to hands-on time for the teachers. The activity was *Pass-It-On*, in which the teachers collaboratively worked on a story project connected to the theme of Halloween (which happened to be on the upcoming weekend). We started the activity by modeling – this enabled us to introduce the basic mechanisms of Scratch (e.g. snapping blocks together, running the program), giving participants what we hoped was enough scaffolding to get started. After the modeling, pairs of teachers had 15 minutes to start their stories. After 15 minutes elapsed, each pair stood up, left their computer, and moved to another computer, where they continued the story that they found at the new computer. After another 15 minutes, the pairs rotated again, and then eventually returned to their original computer to see how the other sets of partners had modified their initial creation. Participants were usually surprised and delighted by the evolution of the project in their absence. (Although some people were sensitive about changes to their original vision!)

We asked participants to talk about their experiences with the activity and how such an activity might work in their own classrooms. One teacher expressed doubt about adding the activity into her lessons. “This was great for me, but I couldn’t let my students get started this way. I’d need to show them more, right? I couldn’t just let them play, right?” She looked around the room at the other teachers for confirmation.

A teacher on the other side of the room quickly jumped in. "I don't think you need to be so structured. I've been using Scratch for about three years. I started using the Scratch Cards with kids because I thought that was a good way to introduce it to them. So I asked them to go through each of the twelve cards before they could start their own project. But that was a big mistake because they got very bored with those cards immediately. Today, what I do with the cards is that I leave them on the table and the kids know the cards are there. They can look for a particular card when they need it. The kids want to be able to just work on their projects and be a little freer."

Another teacher, sitting at the back of the room, shot her arm up, while shaking her head. "I teach it a different way – I don't let them go and do it. Because they just sit there and say, 'I don't know how to make the cat move!' So, I lead them through Scratch step-by-step. It takes me three or four weeks to go through all that. Because if I just ask them to make something, some of the kids – some of them are creative and do produce something – but a lot of them just make something dancing on a screen saying, 'Hi! Hi! Hi! Hi! Oh, you're cool! Hi! Hi!'"

At the center of this discussion is the nature of open-endedness in learning environments. How much freedom and how much structure do teachers need to include in order to create the conditions for students to engage deeply in defining, pursuing, and solving their own problems? The teachers' responses at the workshop represent extremes along a spectrum of open-endedness and structure in activity – and my conversations with teachers revealed a range of thinking across that spectrum. Teachers want to be open-ended in the design of learning activities (as opposed to giving students prescribed problems or problems with pre-defined solutions), but feel the pull of several factors that influence their negotiation of open-endedness in learning: *enabling capacity-building*, *maintaining control*, *perpetuating the status quo*.

Capacity-Building

There are several entry-points to exploring projects in the Scratch online community. You can view projects selected by community members or projects submitted in response to a community-wide activity or projects that the community is currently loving, favoriting, or remixing. One of my favorite entry-points, however, is the listing of "Newest Projects" – all of the projects that are being submitted in real time by creators from around the world. This particular collection of fresh projects has been the method by which I have serendipitously encountered the works of many of my now-favorite Scratchers.

Occasionally, as I have navigated the newest projects, I have encountered a series of nearly-identical projects within a stream of otherwise highly-diverse projects (e.g., Figure 5.1). After some investigation, it was clear that the primary source of these indistinguishable works was classroom settings, where students were being asked to create the same project.

Figure 5.1
A series of nearly-identical projects listed in the "Newest Projects" section of the Scratch website.

Seeing these projects made me think about my early experiences as a computer science major. My first assignment, shown in Figure 5.2, was to type a block of Scheme code (from my lab manual) and execute it.

Figure 5.2
My first programming assignment in college.

```
;;; << Your name goes here >>
;;; Procedure for computing the volume of a sphere.
;;; Inputs:
;;;   radius is the radius of the sphere (must be a number
;;;   greater than or equal to zero to get a sensible result).
;;; Returned value:
;;;   the sphere's volume.
(define sphere-volume
  (lambda (radius)
    (*
      (/ 4.0 3.0)
      (* 3.1415926
        (* radius (* radius radius))))))
```

Not particularly open-ended, in either content or process. But I found the idea of the computer doing these calculations for me fascinating – and I appreciated having a structured introduction to the syntax, which would have been difficult to learn through experimenting with the Scheme interpreter. As the term progressed, some assignments provided opportunities to be more self-directed (less or different structure and constraints), but they were not what I would describe as open-ended. The course was focused on building capacity with programming and core computer science concepts, and the pedagogical approach favored structure over freedom.

The theme of balancing open-endedness and structure in the service of capacity building was recurrent in my conversations with teachers. Teachers talked about wanting to create opportunities for students to explore *and* to build a baseline capacity with Scratch – all while negotiating the constraints of the classroom. Limited time was often cited as a significant concern. Teachers have limited time with students, and students, in turn, have limited time to work on their projects. But design activities, particularly in an open-ended approach, require time.

> You can't just teach Scratch for one day and say, "I covered Scratch." That's the other big issue – you've got to give the kids time to understand it, to play with it, to try to create.
>
> Beverly, Teacher Educator

This constraint led teachers to make decisions about which aspects of a design activity to leave more open-ended and which to make more directed, with student exploration described as less efficient than teacher explanation. For example, Jody shared her strategy of bringing students together as a group for 10 minutes at the beginning of each class to review issues that the students were struggling with, despite student resistance.

> I tell them that they have to give me 10 minutes of their time – and then they can go back to their projects. They would rather just figure it all out themselves, but it would take longer and we don't always have that kind of time.
>
> Jody, High School Computer Science Teacher

Large class/group sizes were also often cited as a significant constraint. Teachers attend to numerous aspects of the learning environment simultaneously, as described by Clive.

> When you're in a lab with 25 kids and they're all pursuing different projects and there's lots and lots of interaction, you have this feeling that you're kind of indispensable both in terms of monitoring behavior and supporting students' learning. And then, being attentive when the school secretary comes through and needs to talk to the students. You feel like your attention is pulled in a lot of directions.
>
> Clive, After-School Mentor

Introducing structure is one way to cope with the scaling complexity as more students are present, as described by Tara.

> I did find that sometimes that being just one instructor, even with six kids, was not enough sometimes. You wanted to be having a conversation with everybody – or everybody was

> stuck and needed something. That would be even harder with a class of 20 or 30 kids. ... If I had a class of 20 or 30 kids, then I'd probably use a little more structure than I was using at the after-school program. I had the luxury of letting them do whatever they wanted.
>
> Tara, After-School Mentor

Adding further complexity, teachers talked about the different levels of expertise and fluency with Scratch – and a different set of ambitions for what they wanted to create – that each student brought. With more open-ended activity, teachers talked about the balance between keeping more-novice students from being intimidated, while simultaneously keeping more-expert students sufficiently challenged. Need varied among students – and within individual students, session-to-session. As Jody described, "Sometimes they need more direction and sometimes they need more freedom."

Structure, direction, scaffolding, and support can be waypoints on a path to greater freedom and self-directed learning. When learning something new, one does not always know what one needs to know, as the student helper of one of the teachers observed.

> The next kid who wants to do a side-scrolling game before they have done anything else – I am going to tell them they need to do some other things first. They haven't done anything yet and they want to jump straight to this and it is not an easy thing to start with. You should do some other things first.
>
> Student Mentor, Working with Clare

And structure can just as easily *undermine* the goals of freedom, self-direction, open-endedness, and creativity. Sadie shared her success with developing a curriculum that had students creating very sophisticated projects that they found interesting and that were developed in very little time, due to a series of step-by-step tutorials and highly structured checklists that she developed for her students to follow. Reflecting on her experience, though, despite its success, she questioned whether she had subverted her goal of fostering creativity.

> It worked out really, really well for them to have this checklist to go through. The one downside about the whole curriculum, looking back on it, is that it was really, really structured. For projects being completed, they were basically told what project to do, and there wasn't too much room for creativity, and I would have liked some other kind of outlet to let them go off and be creative.
>
> Sadie, High School Computer Science Teacher

Control

During my conversation with Arnold, he told me a story about an experience that he had as a student in elementary school music class – an experience that profoundly shaped his philosophy of teaching.

> My mom was a fabulous piano player. I was born with – not my fault – but I was born with perfect pitch. So I was always singing. My sister and I sang duets and all this kind of stuff. And I, in fact, was sent to the principal when I was in fourth grade because I was singing harmony in music class instead of with the others. I was bored and so I was just, you know, trying something different. And I look back on that – what I would have done, as a teacher, is say, "Do the rest of you hear what he is doing? How many want to sing with him?" I mean – it's hard to get fourth graders to sing in harmony! So that's the approach I try to do with my teaching. If a kid does something I don't know about, I try to delight in that and share it with the kids and get the kids excited about it. I think that's the attitude you have to have if you're going to make it teaching something, particularly something you are learning about. That really has affected my teaching because I really believe that you have to be comfortable with kids knowing more than you and not being threatened by it.
>
> Arnold, High School Computer Science Teacher

Arnold's use of the word "threatened" resonated with me – it captured some of the feelings of "nervousness", "fear", and "intimidation" that teachers described to me. Most of these feelings were rooted in concerns about insufficient pedagogical/content knowledge; some teachers felt that they did not know enough about Scratch and about programming to support their students in pursuing open-ended challenges. Selena described this anxiety as an observer, sharing her experiences of supporting a fellow teacher's explorations with Scratch in her classroom and the teacher's lack of confidence.

> I worked a lot with the math and science coach, who was extremely nervous about not feeling like she knew everything about Scratch. You know, like, "What am I supposed to do? I don't know what I'm doing, and I can't teach them how to do it."
>
> Selena, Teacher Educator

Kirby described this anxiety from her position of teaching Scratch for the first time to her high school students. She had very negative experiences with programming as a student in college – yet took on the challenge of teaching Scratch, despite these bad experiences and her fear of not being able to help her students.

> I am not that tech-savvy, and I was really scared about teaching programming because I know nothing about it. I took a course in college because it was part of the education program, and I hated it. I hated everything about it – and I promised myself to never ever take anything like that ever again. In fact, they were thinking about changing the program to having more programming classes, and I said, "No. If they do that, then [*laughing*] I am not going to finish the program!" That's how much I hated it.
>
> The most intimidating thing about programming is that everyone's mind works differently. There's really no one way to do it – there are multiple ways to do things. That's what was hard about helping them with their projects. I would go into a project that was already started, you know, and they're

> thinking one particular way and I'm thinking I'd probably do it another particular way. If I provided my help to them, then it would be in my way. So I had to try to figure out sometimes either "How are they thinking?" and then try and solve from there, or have them ask the right questions so they could solve it themselves which is sometimes really hard for me to do.
>
> Kirby, High School Business Education Teacher

These feelings of fear, anxiety, and intimidation – which are exacerbated by classes with large numbers of students, not having enough time as a teacher to prepare, and feeling isolated more generally from others and support – provoke different pedagogical responses. While some teachers embrace the open-endedness and their unknowingness (which I discuss in greater detail later in this chapter), some teachers opt to moderate open-endedness with greater structure. Structure becomes a mechanism of control over the learning environment – a guard against the potential "chaos" (a popular word in the conversations) of an environment that is too open-ended.

In contrast with using structure to build capacity, which I characterize as the use of structure in the service of making *their students* more comfortable and capable, this other use of structure (introducing structure when negotiating feelings of fear, intimidation, anxiety, etc.) is in the service of making *the teachers themselves* more comfortable and capable. For example, a student teacher anonymously posted to ScratchEd, looking for advice for introducing Scratch in a way that is not "too complicated", aligned with her current level of familiarity with Scratch (Figure 5.3).

Teachers shared similar feelings and pedagogical decisions in our conversations. Taylor, who was now on his second iteration of teaching Scratch, described how his first iteration was much more constrained, with students following teacher-led tutorials.

> In previous years, I would just do a tutorial, like, "OK, we're all going to make this type of project together." It was step-by-step going through the direct instruction part of it, you know,

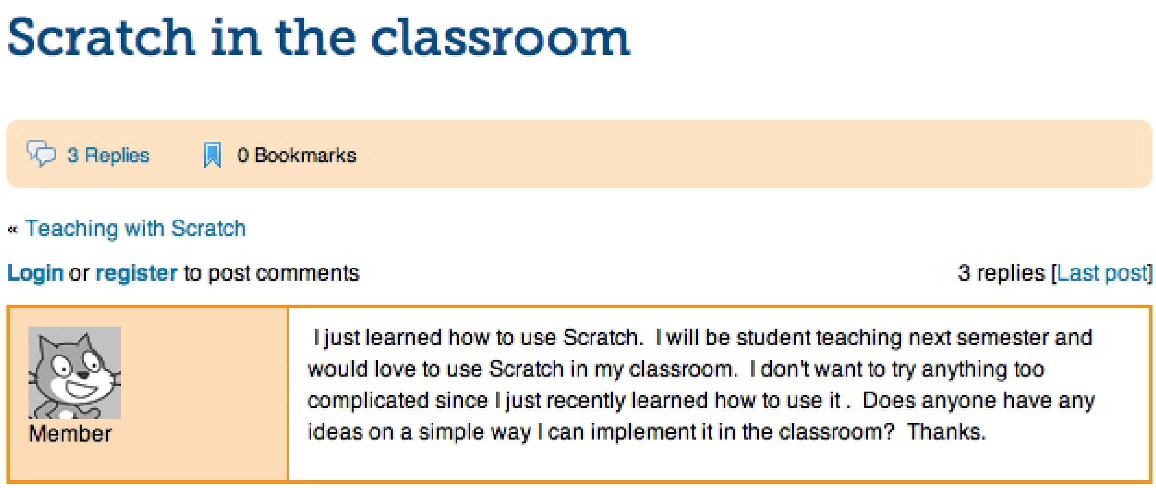


Figure 5.3
A teacher's post on ScratchEd, seeking advice on integrating Scratch in her practice.

> "Do this." They would have the option of what colors to make things or what sprites they wanted to use, but they were all kind of making the same project. It was a very tutorial-type process. Especially since, when I first started using it, I didn't know what I was doing.
>
> Taylor, Elementary School Teacher

Linda described how she started with a more open-ended approach, but migrated to a more constrained approach, based on her own feelings of comfort and confidence.

> I found that when I first started Scratch I said, "Do whatever you want." And then, "Wait a second." It didn't work for me in a way. It was too open-ended. I find that they're going in too many directions for me as a teacher. So I structured the project – it has to have certain things. I never leave it open-ended. It's kind of a teacher control thing. … If you work with the kids, you'll see that they're far more intelligent than I'm making them out to be as far as needing structure. I think it's – as a teacher, being less secure in the environment – I need the structure. The kids would probably move quicker, but it's kind of a security for me. I need to do it this way because that's where I feel comfortable.
>
> Linda, Elementary School Teacher

Whether using a more structured approach or less structured approach, there is still *some* type of opportunity for kids to engage in programming. But Beverly, who helps teachers learn about Scratch, reminded me that fearful teachers may exclude programming activities almost entirely.

> The biggest issue is that they're not comfortable with the material. So they'll cover it so they can check off that they did something, but not to the depth that it's supposed to be covered and not covered so the kids can actually get fluent in it. They'll say, "Oh, the kids created one project. We're done. We did Scratch."
>
> Programming scares the heck out of the teachers. A lot of them just ignore it and teach what they've always taught. That's the fun thing about teachers is that they don't really care what you want them to teach in many cases. They teach whatever they feel comfortable teaching. It's been a real struggle to get them to actually include programming as much as they should.
>
> Beverly, Teacher Educator

Status Quo

Tara is a self-described "radical constructivist" – and in her first experience as a teacher, as a mentor in an elementary after-school technology program that she started at a school in her town, she strived to uphold what she saw as constructivist ideals, giving students considerable space to pursue their interests and work on projects that they care about. In our conversation, she joked about how positively some students responded to her approach in the first few sessions.

> Some of them would just kind of give me this attitude like, "Aren't you going to do this for me?" I'm like, "No, I'm not. I know that you're intimidating and you're taller than me and

> you're very crude – but no, I'm not going to do this for you. You're going to do it for yourself."
>
> Tara, After-School Mentor

Tara was incredibly good-humored about the experience, and she communicated this initial challenge with grace and humility. I thought that this was a fantastically amusing story, but I was certain that it was an outlier, a result of some unspoken happening in the learning environment or due perhaps to this being Tara's first teaching experience.

But this theme of student resistance appeared and reappeared throughout the interviews – across age levels, subject areas, and teacher experience. Students were resisting open-ended approaches to learning because these approaches represent a fundamentally different set of expectations and way of being – opposed to what school culture had otherwise prepared them to do.

> I think some kids are good with exploring and some aren't – and kids should be more like this. Tinkering, just experimenting, going in and putting some blocks in and seeing what happens. Some of my students come into ninth grade and they're not really tinkerers – because they've been taught to do whatever the teacher tells them to do.
>
> Lenore, High School Computer Science Teacher

> For kids, it's such a different experience than how they normally look at school or look at the educational things that they're doing. So when they get stuck, it's "How do I do this? How do I do this? How do I do this?" – and they want the answer from the teacher.
>
> Candace, After-School Mentor

> Some kids are so uneasy with not knowing things, that they need instruction. "Why don't you tell me?" This was one of my biggest things that happened when I first started teaching. I get that much less now just because, I guess, I'm a better teacher. "Why don't you tell me?" "Because I want you to

> learn." But we shouldn't even be engaging in this conversation, you know. It's because you're supposed to – I don't know exactly. It's kids. I don't know if they're trained or teachers in the lower schools just tell them when they ask for all the answers.
>
> Valerie, High School Computer Science Teacher

> A lot of the time the kids want to be told how to do things. After a while, I say, "You know, you're the designer. You've got to figure out what's OK and what details you want." It's often the kids who are reluctant to be creative or reluctant to make a choice. That's because they aren't given opportunities to make mistakes, and they're afraid that if they do make a choice, then I'm going to correct them and say, "Oh, no, no – that's not right."
>
> Georgia, High School Art Teacher

For some teachers, this resistance fueled their determination, as it further underscored a need to prepare students to deal with the open-endedness, ambiguity, and challenge of design activities – both in deciding *what* to work on and *how* to work on it. But for others, student frustration with changes in the approach to learning translated into teacher frustration – and the introduction of greater structure, as a way of perpetuating the status quo of their students' educational experiences and expectations. Crawford, a middle school teacher, described how he had seen this teacher-student cycle of frustration in action, particularly with teachers who were new to Scratch and to more open-ended approaches to learning.

> I think the thing that's frustrating for teachers is, for example, if it's not working, a kid would spend 45 minutes on it and as a teacher you might say, "So what did they learn from that? That they have to persevere? OK, maybe. But they didn't get it – and if you can't solve it by yourself…" There's a frustration level that comes in. With the student saying, "I can't get this to work. This is my idea. How do I do it? I've just spent 20 or 30 minutes or two days on this and it doesn't work!"

> And I think sometimes, when the students are frustrated, the teacher is frustrated.
>
> Crawford, Middle School Teacher

What is a teacher to do when encouraging students to engage in challenging and frustrating activities in a setting that is perceived by students to discourage challenge and frustration?

BUILDING CULTURE

In conversations with teachers, after talking about their motivations, aspirations, and experiences of working with Scratch, I often asked them for advice, as a way to learn more about their implementation strategies. *What did you need in order to get from aspiration to implementation? What would be important for another teacher to know if they wanted to do what you have done?*

Many teachers talked about building a new culture of learning in their classrooms.

> It really helps if you have kind of a culture or climate in your classroom – and not just when you're working with Scratch. It's something you're doing across the entire day. Because it's not just something that you know, "When we get Scratch out, this is the thing we need to do." Or "When we get the computers out, this is the thing we need to do." This is just part of what we do as learners – to help us out, to help us learn and solve our problems. That's my best advice or way that I found makes it work.
>
> Taylor, Elementary School Teacher

Taylor mentioned this idea of "doing something" in this new culture several times, but precisely what was being "done" felt elusive to me. The more I spoke with teachers, however, the more I learned about how they saw themselves and their students in new ways. Five sets of advice about building a new culture of learning recurred in the interviews: *try it yourself*, *follow their interests*, *be a guide*, *feel OK with not knowing*, and *create opportunities to share*.

Try It Yourself

Many teachers remarked that classroom cultural change starts with the teacher exploring the possibilities of Scratch as a part of their teaching practice. Teachers make sense of Scratch by accumulating resources (for example, by searching for materials in the ScratchEd online community) and by hearing testimonials from educators (for example, by attending professional development events or looking online).

But for all of the resources and all of the stories, teachers emphasized the importance of taking the time to explore Scratch themselves, sitting down with a computer and making projects. These hands-on teacher explorations are not motivated by a desire to attain mastery, but to develop a basic familiarity with the interface and to cultivate confidence.

> I think you just need to play with Scratch. That was the big thing for me is that I said, "OK, I'm going to play with this. I'm going to learn this. I'm going to figure out what it can do. I'm going to find some good sample projects that are related directly to education." So, I think it's just not being intimidated by, "Ooh, programming," by that phrase, and just playing with it as much as possible.
>
> Ivy, Elementary Technology Coach

Trying it themselves helps teachers to develop sensibilities around the types of experiences students are having. Clive, for example, talked about how his experiences of working on a Scratch project gave him new insights into his students' processes. As he was working, he noticed that he spent a considerable amount of time thinking about *what* he wanted to create, and then even more time thinking about *how* he would create it – which made him more sensitive in response to his students' quiet moments of "waiting for inspiration" and overcoming challenges.

Follow Their Interests

Larissa, a K-12 teacher and teacher educator, spoke extensively about making space for student interests, using one of the projects her students developed as an example. Larissa asked each of the students in her technology class to select a topic, research it using online and school-library resources, and develop an interactive Scratch project that helps the person interacting with the project learn more about that topic. The topics the students selected were very diverse, including animals, planets, sports, and famous figures in history. One of Larissa's students, Maria, decided to learn all about crocodiles. Maria came to Larissa and said, "I want to know the sound the crocodile makes." After a pause, Larissa responded, "I don't know what the sound of a crocodile is. Let's find out."

In our conversation, Larissa explained the source of her pause – it was a moment of hesitation in balancing Maria's priorities with her own as the teacher.

> If you're this kind of teacher that you like to go with the flow with the students – like with Maria who said, "I want to know the sound of the crocodile" – you realize that you have never, never put any thought into the sound of the crocodile. And maybe if you are a teacher, you are kind of tired because it's an afternoon class, and so you go to the student and tell her, "Well, maybe the sound is not so important; maybe how they breed." But really the student wants to know about the sound. Sometimes, as teachers, we think, "Oh, no, no, that's not important. The important thing is the alligator's claws." But that's not what is important for the kid, so you have to find a middle place in that. For Scratch educators, I would say that imagination is very important, and that the most important part of it is all these interests that the student has.
>
> Larissa, K-12 Teacher and Teacher Educator

Teachers offered numerous strategies for developing interesting activities that connected to students' personal interests. But "follow their interests", as advice, also extended beyond a particular activity, to following students' interests and connecting them to larger socio-cultural contexts. Beau, for example, used his students' interests and passions for video games as motivation for learning about programming.

> Everyone plays video games, you know? This is something that's true now. It wasn't always true, but in this generation, they all play games. Everybody plays video games, like every single person. Why wouldn't you want to learn about that more and be engaged with that more? So that's my starting point. Everyone plays video games. You've all had experiences where you got addicted to a video game. How does that work – and how do you make one?
>
> Beau, High School Computer Science Teacher

Be A Guide

When I was a student teacher, I was amazed by the number of educational aphorisms. Every class offered advice – "friendly, but not friends", "one, two, three – then ask me" – delivered in a cute and pithy package. The phrase repeated most often was to "be a guide on the side, not a sage on the stage". Our program was deeply influenced by constructivism, and use of that phrase aligned with the student-centered approach, reminding student teachers to think of their role differently from that (presumably) of the teachers that we had growing up.

The advice to "be a guide" was common among the teachers interviewed. How could a student learn to solve their own problems, if the teacher was always available as expert, on hand to resolve any challenge that the student encountered? The teachers described their role as guide, mentor, supporter, facilitator – helping students to pursue their goals through metacognitive support, by asking questions, providing helpful resources, breaking down problems into smaller problems, and reframing problems.

Linda shared her experiences of being a cognitive guide, relating a story about one of her students' experiences with Scratch. Her third-grade students were studying animals that live in the water, and students were invited to represent their investigations in whatever format they liked. One of her students, Sarah, decided to create a Scratch project that showed the different types of environments that fish live in. But Sarah quickly got stuck as she worked on her first tableau, that of a fish in a pond. She wanted her fish to swim back and forth in the pond, but she was incredibly frustrated because no matter what she tried, her fish was sometimes swimming backward (Figure 5.4). Deeply frustrated, she thought about abandoning her project idea, but decided instead to call Linda for assistance.

Figure 5.4
Buggy Scratch project in which the fish does not always point in the direction that it is swimming.

Linda sat down, and asked Sarah to tell her about the project and the bug that she was working on. Sarah struggled to describe the problem, but after running the program a few times, she found the language to describe the situation – that her sprite was pointing in the wrong direction. Sarah experimented with a few blocks, but the project still wasn't working as she imagined.

Sarah had done what Linda would have tried, so she wasn't sure what the solution to the problem might be. Linda suggested a next step. "We know that there's a way of doing this – because we've seen examples. Why don't you look for similar projects?" She could tell Sarah was unhappy with the project not working correctly, even during this period of research. "And until you figure out direction," Linda suggested, "you could change your fish to another animal that lives in the water that doesn't have a direction." They brainstormed a list of aquatic creatures that might move laterally, and Sarah temporarily revised her project to feature a jellyfish (Figure 5.5).

Teachers also talked about being emotional guides, helping students negotiate the emotional landscape of autonomy, challenge, experimentation, and creativity. Teachers encouraged students to be fearless in their experiments.

Don't be afraid to just try things, to see what happens. It's not the end of the world.

Clare, After-School Mentor

You can try it and it's OK. The world won't end if you try to do something another way. If you're not afraid to tinker, you're better at it.

Valerie, High School Computer Science Teacher

You can't break it, so why not? If you're curious about something, try it, see what happens. It's not like it's going to be broken forever. We can go back to where you were before. But you know – take a chance, take a risk, see what happens!

Sabine, Elementary School Teacher

Teachers also encouraged students to persevere.

It starts on the first day of school – getting kids to appreciate that they're going to make mistakes and that I'm going to be asking them to do stuff that is hard. I always just put that right out there. And they don't, at first, just because they want to succeed – and part of it is the age, too. Well, even adults don't like to fail, or make mistakes. But it is important, I feel that when you do run into difficulties that it's not time to give up or cry. It's time to think about the strategies that you have to solve your problem, or to look for help. I hope they get to appreciate it by the time they leave my class. I do see less frustration as the year goes on. That could be them maturing or, you know, I hope it has something to do with me just encouraging them and continuing, honestly, to put them in situations where the problems are hard. You know? And that they kind of get used to that. No reason to break down or give up – you keep at it.

Taylor, Elementary School Teacher

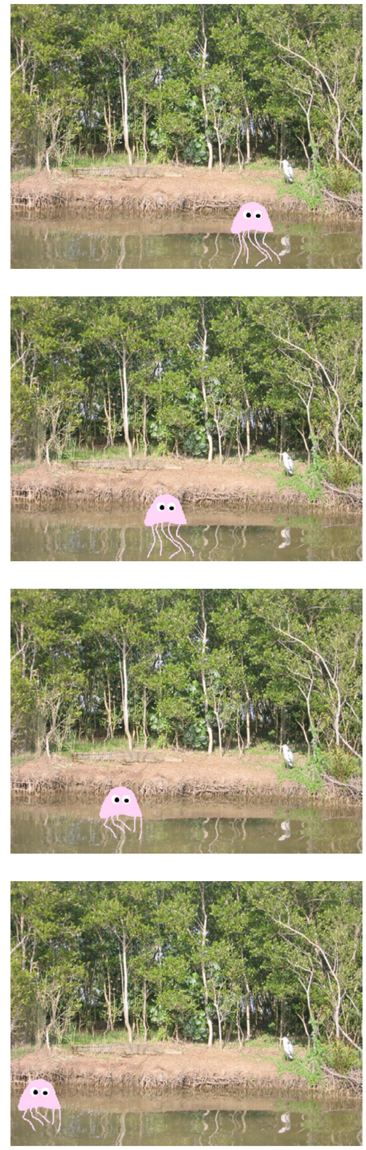


Figure 5.5
Scratch project revised, substituting the fish for a jellyfish, as a temporary simplification of the animation.

Feel OK With Not Knowing

Returning to the aphorism about the "guide" and the "sage", teachers talked about what it would even *mean* to be a Scratch "sage", what it would mean to "know" Scratch. All of the teachers that I spoke with felt that it meant knowing that you can't know everything – and that it was important to feel comfortable with this realization.

> You need to be able to divorce yourself from being the expert. You're not going to say, "All right, I know everything there is to know about Scratch," because that's impossible.
>
> Tara, After-School Mentor

Comfort and confidence are developed not by mastering the universe of answers to every problem. Rather, they are supported by having a set of problem-solving and design strategies in place, and having a particular psychological stance. Scratch and programming aren't about knowing an answer, but about flexibility in one's thinking.

> I think the really nice thing about programming is how one thinks, that you approach a problem in a kind of an amorphous way. You don't have to come into a problem scared if you don't know everything. You approach a problem saying, "I can approach a problem without knowing everything." I can say, "I'm going to solve this like this, and I'm going to assume certain things are going to fall into place, and then if they don't I'm going to adjust things."
>
> Valerie, High School Computer Science Teacher

Further, it involves openness to the curiosities of the experience.

> The biggest thing we've realized about teachers is that teachers have a mindset of needing to know everything before they're willing to offer it to students. As if they need to know everything about programming Scratch before they can offer Scratch. And the power behind Scratch is that you actually don't need to do that if you have accessible materials and

> kind of a playful community. … [Adults] seem to be the ones who are having all the trouble feeling comfortable in being able to say, "I don't know. Let's find out." You'll hear that mantra in most of our programs. We don't expect people to be experts. In fact, we expect people just to be curious and take the stance of a learner.
>
> Kent, Teacher Educator

This opens up opportunities to disrupt the teacher's role as "the one who knows". In Mindstorms, Seymour Papert tells the story of a teacher who was working with a student on fixing a bug.

> As they puzzled together the child had a revelation: "Do you mean," he said, "that you really don't know how to fix it?" The child did not yet know how to say it, but what had been revealed to him was that he and the teacher had been engaged together in a research project. The incident is poignant. It speaks of all the times this child entered into teachers' games of "let's do that together" all the while knowing that the collaboration was a fiction. Discovery cannot be a setup; invention cannot be scheduled. (1980, p. 115)

The teachers that I spoke with related similar stories of struggling over authentic problems with their students, and they shared the power, surprise, and delight experienced by both the student *and* the teacher in doing so. For many teachers, this delight was expanded when they weren't just learning *with* students, but *from* students. Linda, for example, talked about the pleasure she experienced by learning from students, by having them explain things to her and by studying their code. Clive described how being open to learning from students enhanced both his own learning and the students' learning.

Create Opportunities To Share

Jody was initially very skeptical about the idea of her students remixing each other's work. She wasn't sure how her students would react to other people looking at and changing their code – or of what her students would actually be learning from the experience. At a meetup with other Scratch educators, Jody had the opportunity to hear about another educator's use of remixing in her classroom, a type of pass-it-on activity. Jody felt inspired by the activity and decided to try it in her own classroom the next week.

At the following month's meetup, Jody's stance toward remixing had shifted significantly, and she shared her experiences and her enthusiasm for remixing, as a practice, with the group. In conversation, she elaborated on the value that she saw in creating opportunities for students to share with each other.

> I like to see them share. The difficulty is that that's not what they're used to. Maybe they're used to sharing their homework at night and all that, but they're not used to coming into class and finding out something and being asked to let everybody else know that very cool thing. It kind of goes against their whole view of what school does sometimes. It's a learning curve from their standpoint.
>
> What we do is – they post their programs online, and then other people play it, comment on it, remix it, or fix it. It's very hard for them to admit that something doesn't work, but they'll say, "Here it is. It's a little glitchy. I don't know why." And then somebody else will pick up on it and look at it and try to fix it. Which is great because in the end, it's a win-win situation. I don't think it's all that healthy for them to feel like they have to do the whole thing on their own and figure it out all on their own. I mean, that's not the way we all work, so why should they?
>
> Jody, High School Computer Science Teacher

Teachers talked extensively about the importance of sharing – at different stages of project creation, with different numbers of people, over different periods of time. Kirby talked about the importance of *presenting* final work, by having kids participate in a gallery walk, playing each other's games and asking each other questions. Taylor described the shift in kids' thinking through *collaborating*, creating interactive narratives with a partner. Clare talked about *demoing* as a way of supporting peer learning, with students starting each class session sharing something they had figured out in the previous class. Whatever the form of peer sharing, teachers valued it as a powerful and critical element of learning. As Sun described,

> The students wanted to learn more. They used to work one-by-one, but then they would encounter a big mountain or big barrier. That is why I encourage them to make a community – without community, it's not possible. The tool is important, but making community is more important for learning.
>
> Sun, Elementary and College Teacher

LEGITIMIZING LEARNING

In our conversations, teachers described the excitement they felt about the learning environments they were designing and the learning they were seeing in those environments. But an uncertainty often accompanied the excitement, as teachers described struggles in reconciling their classroom activities with the expectations and structures common in formal learning environments.

Two fundamental aspects of Scratch were most often questioned (either externally, by administrators, colleagues, or parents, or internally, by the teacher him/herself): *social/networked learning activities*, and *assessing learning of creativity and problem-solving*.

Networked Learning

Although some teachers that I spoke with did not have reliable access to computers (e.g., considerable demand for shared computer labs limited the time available with computers) or network infrastructure (e.g., low-to-no connectivity in rural, or older urban, schools), most teachers were in settings in which there were computers and network connections. Some of these connected teachers described the Scratch online community as an important resource, both as a source of help and as a way of learning how to participate in network culture. But it was a resource that was described as having great complexity in the school environment.

School models of learning are deeply individual. Teachers often described a tension between wanting to support social learning activities, such as pair programming and group design, but also wanting to assess an individual student's understanding and development over time. Scratch and its networked learning approach, with access to hundreds of thousands of other young learners and millions of projects, only amplifies anxieties about assessment. We see these anxieties expressed through numerous questions on the ScratchEd forums, and through emails (Figure 5.6).

Figure 5.6
Teacher asking about copied work, in an email to the ScratchEd Team.

Subject: **Checking for remixing**
Date: December 3, 2011 5:35:19 PM EST
From: --------------------------
To: scratched@scratch.mit.edu

Hi, Is there a way for educators to simply check student work to determine how much is original, and how much was simply copied from the Scratch site?
I don't want to upload individual projects from my students. I just want a way to see if the work is original or not. Thanks.

Teachers who encourage their students to engage with the online community described their approaches to supporting students' thinking about using others' work. They emphasize the importance of credit with their students, that it is important for students to acknowledge their sources. Some teachers define rules about when it is appropriate to use others' work. Jody, for example, encourages her students to experiment with implementing their own ideas, rather than finding a similar base-project to build on.

Concerns about authorship and copying are accompanied by concerns about student safety, part of a larger cultural conversation about kids' appropriate engagement with online resources, applications, and communities. Teachers face questions from administrators.

> Our school and district is big on this PII (Personal Identifiable Information) concerns. Does anyone else have these issues with their school or district? If so, how were you able to overcome them? … If any of you have an idea on how I can try to ease the concerns of our district over these concerns.
>
> Excerpt from ScratchEd post

They also face questions from fellow teachers.

> There are teachers at my school who are nervous about the idea of kids contacting "strangers" on the Scratch web site.
>
> Excerpt from ScratchEd post

Parents were also a source of concern when thinking about encouraging students (both younger students and older students) to connect to the online community. These concerns led teachers to restrict the types of interactions their students had in the online community –

> I usually say [to the students], "We're going to share our projects with the rest of the world" – even if I sometimes don't allow comments because I have to be careful with my parent body.
>
> Linda, Elementary School Teacher

Sometimes the concerns led to not using the online community at all.

> I don't. I don't. That's something that they have to find themselves. If I was privately tutoring them I surely would. In the class, without their parents' permission, I'm not comfortable doing that. If it's outside of school, I am. But when it's a school thing, I'm really not supposed to do that. I don't know if the parents want them uploading, downloading, looking at games, or what. I'm sure this is ridiculous, because they

look at everything and anything, but I'm not supposed to. So I really haven't.

Valerie, High School Computer Science Teacher

But for some teachers, these concerns about copying and safety never reach their classrooms, as school or district restrictions prohibit the use of networked tools like the Scratch online community, categorizing it as a "social networking platform" like Facebook or a "media file-sharing site" like YouTube. In a survey of Scratch educators conducted in May 2012, 468 respondents indicated that they were enthusiastic about the learning potential of the online community, but 68 respondents (14%) were constrained to using only the authoring environment, as the online community did not conform to district policies.

These types of restrictions are frustrating for teachers – and for students. A high-school student's Scratch project (Figure 5.7) about her experiences of using her school's library computers catalyzed Scratchers to share their experiences and frustrations with school network blocks.

Figure 5.7
High-school student's Scratch project about her experiences with school computer networks.

These frustrations lead kids to develop creative workarounds to the restrictions. Tara, for example, shared how two of her students dealt with her school's network restrictions.

In the first session of her after-school Scratch club, Tara showed her students a few of the sample projects that come with Scratch. Brian and Ben were immediately inspired, and had an idea for a Club Penguin-inspired project they wanted to create. In their vision, a penguin would roam around a small town and would be able to go into different buildings in the town – a building to find food, a building to play a game, a building to watch a movie, etc. But they weren't sure how to start, and they couldn't find anything like it among the sample projects. They asked Tara if she had any ideas. Tara suggested that they explore the Scratch website, but all three were disappointed when the school's access policy page appeared in the web browser.

At the next meeting, Brian ran in and threw himself next to Ben at the computer. When Tara asked what all of the excitement was about, Brian explained that he had found the Scratch website at home – and had found a project that partly did what he wanted it to do. He had downloaded the project and played with the code until it did what he and Ben were trying to do with their project. He had memorized the chunks of code at home and was now trying to recreate the effect in their project at school.

Tara watched Brian and Ben enthusiastically implement this method of memorizing chunks of code and carrying them between home and school for several weeks. Every session they would tell Tara, "I did the coolest thing. Do you want to see?" And then they would try to show her. As the project became increasingly complicated, they would have more trouble with the approach, telling Tara, "I don't know – it works at my house. It worked!" Tara decided to buy the students USB drives to make it easier for all of the students to bring in projects or excerpts of code that would help them with their own projects.

Creative Learning

When talking with teachers about how they know what young people are learning with Scratch, I often hear, "I know it when I see it" – with teachers relying on their deep intuitions about student learning. For example, teachers would relate stories of their students' persistence, dedication, and enthusiasm for the work, as possible indicators of learning.

> They love it. I can't emphasize that enough. I heard one boy with his mother – his mother is a teacher at the school – and as they were leaving, I heard him in the hallway say, "Can we do Scratch this weekend?" And she said, "You've got a ball game this weekend." He goes, "I'd rather do Scratch." That said a lot for a little 7-year-old boy, that he was more interested in doing Scratch. I had just handed out a new program for them to do, and he was taking his copy home and he wanted to work on it over the weekend.
>
> Sabine, Elementary School Teacher

But the teachers talked about the stress they face when lacking language or other (often, measurable) structures to explain to those outside of the learning community the complexity of the learning taking place in their classrooms. They shared how this stress leads teachers to search for methods of assessment that will demonstrate the legitimacy of their work. Unfortunately, teachers often described themselves as being in one of two uncomfortable situations.

In the first situation, teachers felt unable to assess, and, as result, omitted assessment altogether. The inability to assess was often rooted in concerns about poor assessment models potentially undermining their teaching goals, by undervaluing multiple approaches to learning and by constraining creativity.

> Assessment is continuously a challenge. I kind of have a gut feeling that they're progressing of some sort. But unless I make the criteria or develop a rubric I can't tell really how they're doing. I do see growth, but I don't have a real clear-cut

> way of assessing. But I think a rubric would probably be the way to go with trying to determine what is it that I wanted to see and where do they come within that range. But it's hard because – how do you put a rubric on creativity? Sometimes if they come up with some really good idea, but it's really hard – I haven't figured that out. So, I have the advantage of being a little open-ended with assessment. … I don't know how you assess the ability to think logically except that maybe – I have no idea.
>
> Linda, Elementary School Teacher

> I never ended up developing anything [for assessing]. I kind of just did it more along a pass/fail basis or a complete/incomplete because I didn't – I didn't want to be the one responsible for breaking their creativity either. … I didn't want programming to feel like an assignment or work. Or something that was out of reach for them. I wanted them to have fun with it and be creative with it. More structured assessments would have looked a lot better on paper, and then we would have been able to mark off our objectives at the end of the quarter and say, "OK – this is what we accomplished." But, instead, we kind of just have soft objectives that we can just sort of look back and say, "I learned this and that," nothing measurable. I guess, I don't have anything measurable is what I regret.
>
> Kirby, High School Business Education Teacher

In the second situation, teachers felt able to assess, but in a way that was not connected with their overarching goals and aspirations. Jody, for example, described how she used small programming challenges to assess students' technical understandings, although it came at the cost of creative diversity.

> The assessment question is a huge question. … I understand that in computer programming courses even in college there's so much plagiarism because there's typically one well-written answer and that's the answer, and I wouldn't want that to be

> a part of the learning of Scratch or the learning of programming in my class. And so you really need to keep coming up with ways of making it – so even if it's a structured assignment – how do you make the end products look different? How do you add some personality?
>
> Jody, High School Computer Science Teacher

She balanced this more-constrained assessment through programming challenges with an assessment of effort in the final projects.

> It's basically assessing the efforts that you put into these things, rather than if it didn't work. In their final projects, hardly any of the projects worked to begin with, because the students took on big ideas. Part of it is to throw them into that environment, where they realize that it's a lot harder than it looks. And then they can't back out because they're so tunnel-visioned into it, and that's really difficult. … It's a real struggle. So, I wouldn't want to mark them down for it not working, as long as they kept plugging away and fixing and fixing and fixing, and I'm there assessing it. To be honest, on a daily basis, I'm assessing their effort.
>
> Jody, High School Computer Science Teacher

Teachers talked about how assessments would help "market", "sell", or "prove" Scratch, to "convince" colleagues, administrators, and parents about its validity. This was particularly important, Crawford noted, when doing something different in the classroom.

> I think where I'm struggling with Scratch – I want them to be creative, but – and this is another dilemma that is interesting – is I feel this pressure. This isn't a pressure from anything anyone has said explicitly, but it's this undercurrent of education that if you deviate from the curriculum, you still have to prove it works, even more so.
>
> I was in parent conferences and we'd done Scratch in my math and science class. I framed it as, "Now we're working on computational thinking, computer science, and science."

And then it sounds fancy versus "playing video games". It's this pressure of – "Well, this class is doing this." "Why are you doing *this*? What skills are they learning?" And sometimes I feel that pressure not from anything explicit, but it's from society – and it's interesting.

Crawford, Middle School Teacher

Chapter 6

AGENCY/STRUCTURE

In this chapter, I focus on the relationship between agency and structure, as manifested in the learning environments described in Chapter 4 and Chapter 5. For each setting, I identify the sources of structure encountered, accessed, and adapted, and discuss how those structures enabled or constrained the agency and activities of kids and teachers.

In Chapter 4 and Chapter 5, I described the experiences of kids in the Scratch online community and of teachers working with Scratch in K-12 classrooms. These chapters serve as a response to my first research question – *How do out-of-school and in-school learning environments support the activities of computational creators?*

In this chapter, I focus on the relationship between *agency* (defined earlier as *a learner's ability to define and pursue learning goals*) and *structure* (defined earlier as *rules, roles, and resources, both explicit and assumed*), as manifested in these two learning environments, by identifying key sources of structure and describing how particular structures enable or constrain learner agency. In doing so, and drawing upon the data and analysis from Chapter 4 and Chapter 5, I respond to my second research question – *Within out-of-school and in-school learning environments, how does structure enable, rather than constrain, the agency of young computational creators?*

I also describe the ways in which structures made available by the Scratch and ScratchEd teams have been adapted in ways that enable or constrain learner agency. As de Certeau (1984) argued, one must think about the ways in which people will adapt and interpret structures –

> It is nonetheless implicit in the "producers'" claim to "inform" the population, that is, to "give form" to social practices. … To assume that is to misunderstand the act of "consumption." This misunderstanding assumes that "assimilating" necessarily means "becoming similar to" what one absorbs, and not "making something similar" to what one is, making it one's own, appropriating or reappropriating it. (p. 166)

Adaptations of Scratch- and ScratchEd-developed structures have taken on several forms, but, as connected to my goals as a designer (described in Chapter 2), have sometimes manifested as incongruent adaptations (Lin & Fishman, 2009) or "lethal mutations" (Brown & Campione, 1996) that undermine agency and/or constructionism in learning environments.

This chapter is organized into two parts. The first part – *In The Wild* – revisits the experiences of kids working with Scratch at home. I describe the structures encountered by kids in the Scratch online community, from the perspective of these young, primarily self-managing learners. The second part – *In Classrooms* – revisits the experiences of teachers working with Scratch at school. I describe the structures encountered by kids and teachers in K-12 classrooms, from the perspective of the teachers who are responsible for designing these learning environments. For each setting, I identify the sources of structure encountered, accessed, and adapted, and discuss how those structures enabled or constrained the agency and activities of kids and teachers.

IN THE WILD

In Chapter 4, I focused on the goals and ambitions of young creators in the Scratch online community. Kids described how they valued the freedom to create and learn, and the potential to connect with others for help and as audience. They shared their challenges, and their strategies for overcoming those challenges. In general, the conversations (and Chapter 4) were organized around what kids want and what kids do – and what helps them and what hinders them.

Table 6.1 summarizes the main sources of structure that the kids discussed in our conversations – elements characterized as important to (or as interfering with) their goals – and how these structures enable or constrain learner agency. In the remainder of this section, I discuss each of these four sources of structure: *individual*, *computer*, *home*, *school*.

Individual: Personal Interests And Current Abilities

For kids in the online community, their personal interests, passions, and curiosities served as the primary enabling structure of their creative work. Kids give themselves the freedom to pursue their interests, whether exploring particular topics, or particular features of Scratch, or particular genres of projects. In addition to serving as creative inspiration, connecting to personal interests is a significant

Examples
In *Enjoying Freedom* (Ch. 4), Lana's range of creations, Jackson's pursuit of "peaks", and the varied sources of creative inspiration kids draw on.

Table 6.1 Sources of structure available and experienced at home, participating in the online community.

Source	Structure	Enable(s)	Constrain(s)
Individual	Personal Interests	By serving as a source of creative inspiration; By providing motivation to persevere	When leading to goals that exceed current abilities
	Current Abilities	By increasing creative confidence; When appropriately matched to goals	When insufficient to achieve goals
Computer	Authoring Environment	By providing easy entry-point; By supporting a range and diversity of creative possibility	By providing too little direction; By lacking complex features
	Online Community	By providing ideas and technical/aesthetic building blocks through enormous repository of projects; By providing access to other designers for ideas, advice, collaborative work	When failing to connect learners with projects and people of interest, particularly as the community grows in size and sophistication; When peer pedagogical knowledge or content knowledge insufficient
Home	Parents, Relatives, Siblings, Friends	By providing emotional, technical, and metacognitive support	When imposing direction without latitude for self-discovery; By specifying computer-use/screen-time restrictions
School	Teachers	By introducing kids to Scratch or other learning opportunities	By designing low-agency learning experiences, with programming or other topics
	School Culture	When connecting out-of-school Scratch learning and school-based learning	By offering learning experiences perceived as fundamentally different from "real life"

source of kids' motivation to persist on hard problems. When projects are challenging, kids rely on their interest in the work as a way to continue making progress.

Examples
In *Getting Stuck* (Ch. 4), Edgar's side-scrolling challenge, Adele's infinite pylons, and Connor's "Later" folder.

Personal interests constrain learner agency, however, when they lead to goals or ambitions that exceed the creator's current abilities. *Abilities* is intended broadly here, including technical, emotional, cognitive, and metacognitive capacities, strategies, and dispositions that kids draw on while creating projects. There were numerous stories of mismatch between interests and abilities – often with kids who were newer to Scratch or who were younger. Kids described wanting to create projects, but being unsure about how to use a particular feature, how to organize a particular set of blocks, or even what the barrier was. As 10-year-old Robin noted, "You don't know what you don't know until you know it."

Certainly, a lack of abilities constrains kids' participation and learning, particularly when kids have not yet developed strategies for seeking help. But, more generally, kids' abilities (whether knowing how to use Scratch or how to break down problems or how to find support outside of themselves) serve as a significant enabling structure. Kids described how they built on their existing expertise and abilities to develop and improve over time. 9-year-old David, for example, who had hundreds of projects, could see how his increasing abilities had expanded the range of what he was able to create.

> My very first project doesn't look as interesting to me, but I'm still surprised at what I did back then.
>
> *Why don't you find it as interesting?*
>
> I don't know. It seems like now I feel like I can do bigger things, now that I know a little better.
>
> David, 9 years old

Several kids described how their abilities had developed to the point where they felt completely unconstrained in what they could create. They were still motivated by curiosity and creativity, but the challenge was no longer about figuring out the interface or developing strategies for getting unstuck. For example, 13-year-old Sebastian shared one of his most recent projects, a complex game based on a popular phone application, and I asked him what he had to learn in order to complete the game.

Examples

In *Making Progress* (Ch. 4), the range of strategies that kids developed, including experimenting (Jenson's Bananagrams analogy), planning (Lana's discipline increasing with complexity), compromising (the "Birds" version of "Angry Birds"), persevering (Eva's work on a much-loved project), and taking breaks (David's Scratch workbox).

Was there anything you didn't know already for Fruit Ninja that you had to learn?

I mean, I guess I'm at the point now where I'm not learning anything new. I'm just using it in different ways.

Sebastian, 13 years old

Computer: Authoring Environment And Online Community

The Scratch authoring environment was designed for young people with no prior experience – or even prior image of themselves – as programmers. But although it was designed to be very easy to get started with programming, Scratch was also intended to support kids' ever-expanding understandings and abilities, supporting them in creating more complex projects over time. The range of projects in the online community, from first steps to technical masterpieces, and the range of projects among the kids that I interviewed serve as evidence that the authoring environment enables the agency of both novice and more-expert creators.

However, in striking a balance between the needs of both novice and more-expert creators, the authoring environment, in some ways, compromises the needs of both groups. Although Scratch provides some support for getting started (as discussed in Chapter 2), this support is insufficient for some kids, which necessitates finding other sources of support. As we saw in Chapter 4, this has provoked kids to develop new resources, such as Scratch tutorials and the Scratch wiki (the online compilation of articles about Scratch, for Scratchers, by Scratchers). 12-year-old Clark, who had actively contributed to the Scratch wiki, described his initial confusion with Scratch, which motivated his contributions to the wiki.

Examples

In *Making Progress* (Ch. 4), kids developing resources, like Nevin making tutorials, Clark contributing to the Scratcher wiki, and Sonia making math projects.

At first, [Scratch] was pretty much a foreign world. And definitely, if I had the Scratch wiki back then, that would have helped.

Clark, 12 years old

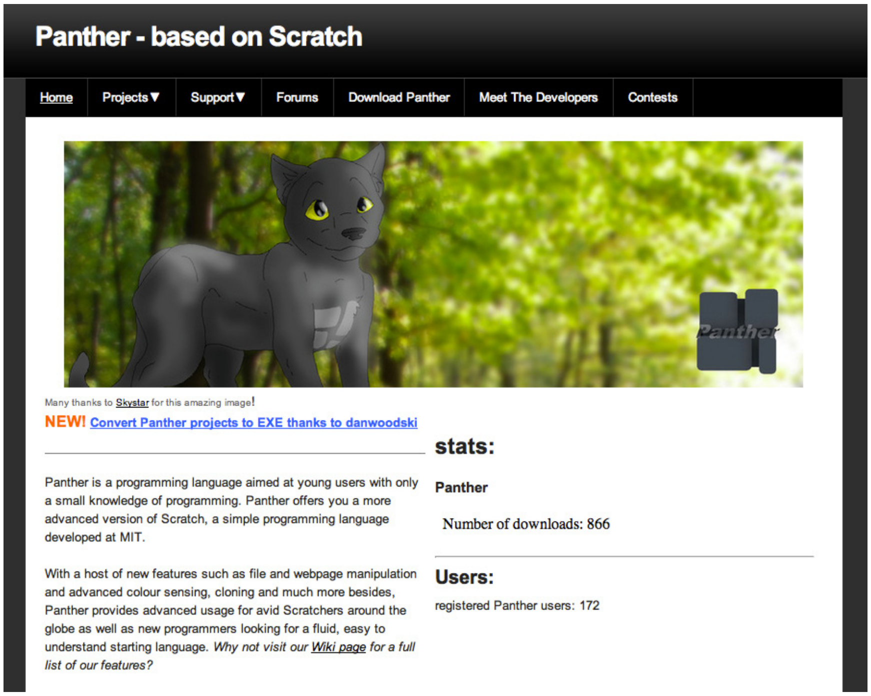


Figure 6.1
Panther, a derivative version of the Scratch authoring environment.

For other kids, Scratch – in its current incarnation – is not sufficiently complex. An unexpected adaptation of the authoring environment came in the form of young people downloading the source code and developing their own versions of Scratch. For example, a group of kids developed a derivative version, named Panther, which includes additional blocks and features (Figure 6.1).

The Scratch online community is also an enabling structure for young people's development and agency as computational creators. It provides young creators with access to a large collection of projects and to other creators. Having access to projects provides kids with ideas and guidance for the projects they want to create. As the site has grown, however, it has become more challenging to find the content that is most relevant or most interesting. This can sometimes lead to a feeling that everything has been done, as Lana described.

Examples
In *Making Progress* (Ch. 4), Bradley learning from others' trig projects.

> When Scratch was first introduced, there were a lot of things to discover about Scratch. And people were really enthusiastic about it. Now, a lot of people are like, "Oh, we miss 2007. Things were so good in 2007." That's because everybody

> shared a common interest – Scratch and getting to know more about Scratch. Because not a lot was known, not a lot of things were known about it. So, once this knowledge kind of accumulated and a lot less things were left to be discovered in Scratch, eventually there came a point when basically, most of the things in Scratch were kind of common sense in the community. Everybody knew about everything.
>
> Lana, 17 years old

Examples
In *Making Progress* (Ch. 4), the utility (and challenge) of forum exchanges, Aaron learning about absolute value, Eva's anti-remixing sentiment.

In *Finding Audience* (Ch. 4), Brent experiencing mean comments, Aaron's project about fame frustration, Chuck dealing with too much attention, the call for a ban on cat-themed projects, earnest and cynical advice projects about popularity.

In addition to having access to projects, having access to other creators enables kids' creative goals. Other creators can provide feedback and advice, and serve as audience or as collaborators. And just as the increasing size of the site can make it difficult to find interesting projects, it can present challenges in connecting with others. As we saw with some of the quantitative and qualitative data in Chapter 4, kids often struggled with finding the attention and connections they desired. Further, the kids were sometimes constrained by peer interactions that were not as respectful (kids testing social boundaries, or learning how to interact with others online) or as helpful (due, perhaps, to a lack of pedagogical or content knowledge) as they – or the Scratch Team – might hope.

Home: Parents, Relatives, Siblings, And Friends

Examples
In Table 3.1, kids introduced to Scratch by family and friends.

From the first interview, it was clear that kids' home lives served as an important enabling structure in their development as computational creators. Parents, relatives, siblings, and friends were often mentioned in interviews – and sometimes participated in the interviews as well. (It was these supportive home lives that served as the initial prompt for me to think about Scratch in schools, as a form of support for kids who did not have the benefit of Scratch support at home.)

Examples
In *Making Progress* (Ch. 4), parents and others at home supporting computational creation.

Parents (and other home-based supporters), whether they had technical expertise or not, were often unfamiliar with Scratch itself. They contributed by supporting kids technically (e.g., working collaboratively to figure out some aspect of Scratch), emotionally (e.g., serving as an appreciative audience), and metacognitively (e.g., helping break down a challenging aspect of a project). 10-year-old Evan's father, for example, shared his thoughts via email after the interview. He had been listening in to the conversation and wanted to share his perspective.

> I help Evan very little on his projects, mostly with metaphysical stuff (how to upload projects was the last thing I helped with, and will help later this week with how to move sound files from Garageband to Scratch). Every other time, I just come when he calls me. He asks me questions which I don't answer ('I don't know' is my typical response), and he pretty much figures things out himself.
>
> Email from Evan's father

Home-based supporters can also constrain agency. Some of this happens when technical expertise leads to "overhelping", as we saw in Chapter 4 with Chuck, who felt at times that he was "teachered" by his mother, being told the "best way" to do things rather than independently discovering and exploring. Other constraints that emerged from the home setting were restrictions on time or access to computers. Some parents, for example, had strict guidelines about the amount of time that their kids were permitted to use the computer and to use Scratch.

School: Teachers And School Culture

Schools were a more peripheral part of my conversations with kids, but not absent and certainly providing structure for young creators, even in the home environment. Although two of the kids that I spoke with had been homeschooled, the majority attended public elementary, middle, or high schools. Their experiences with computers at school varied considerably, from kids who did not have access to computers at school to kids who were encouraged and expected to use computers as part of their daily activities. Some kids had learned about Scratch at school, and those that had were very appreciative of the support that they had received from the teacher who had introduced it to them. For example, 14-year-old Ashleigh was introduced to Scratch by her computing teacher as something to try

Examples
In Table 3.1, kids made aware of Scratch by teachers.

independently, and was encouraged to enter a video-game-making competition, which she won. Although she worked with Scratch at home, she might not have found this opportunity without school-based support.

Conversely, kids described how Scratch enabled their school learning. Kids talked about using Scratch for school projects (such as developing book reports, exploring math concepts, creating science simulations) and how the types of thinking that they were learning also became a part of their cognitive toolkit. For example, 11-year-old Jenson and his mother reflected on his improving mathematical ability.

> We don't do Scratch at school. My dad just got Scratch up and I started doing Scratch at my house. I do it every once in a while, like mostly like every two days because – I usually only get an hour to play any other game, but my dad lets me get on as much as I want on Scratch because it's a learning type thing, so I get on it very often.
>
> *What does that mean, "a learning type thing"?*
>
> Well, actually, my parents think that's what got my math up higher, like problem solving in math, and stuff like that.
>
> *And what do you think of that?*
>
> I think that's definitely a part of it. It definitely encouraged me to get more into math and stuff like that.
>
> (Mom) Do you want me to speak to that?
>
> *Sure!*
>
> (Mom) About half way through the year or so, his teacher came to me and said they were kind of stunned with the way his math skills shot up. We kind of point to the problem solving that he's using in Scratch. He has been identified as TAG (Talented and Gifted) now – and that has just not been the case. It really has to do with his math scores, and his math scores shot up just this year. You know, it could do with a lot

of things, but we are very happy about the Scratch contributions to that.

Jenson, 11 years old, and his mother

At the same time, however, kids described how they felt constrained by the structure of school. Some of the constraints were specifically about their Scratch participation. For example, 13-year-old Eva talked about how restricted access to computers at school made it difficult for her to introduce her school friends to Scratch. But many of the constraints were about school culture more generally. School learning was described as separate from "real life" or kids' "real interests". For example, 10-year-old Easton is an incredibly passionate, self-directed learner, having taught himself electronics and puppet-making and a variety of other creation-oriented activities by watching videos, but feels disconnected from school (which his mother confirmed).

Examples

In *Enjoying Freedom* (Ch. 4), Lori's use of code exactly as written on the board by her teacher.

You know what? I'm inspired by YouTube.

(Mom) Oh, yes, YouTube videos.

They have really cool things on YouTube.

(Mom) The things that excite him don't really happen at school I think.

I hate school.

Easton, 10 years old, and his mother

IN CLASSROOMS

In Chapter 5, I focused on the experiences of teachers introducing Scratch in K-12 classrooms. Teachers shared their ambitions for supporting their students as creative problem solvers, anxieties about the open-ended nature of design activities with Scratch, strategies for coping with challenges, and struggles with legitimizing the learning taking place in their classrooms. In this section, I identify and analyze, from a teacher perspective, the structures in K-12 classrooms that enable or constrain learner agency.

Table 6.2 summarizes the main sources of structure that teachers discussed in our conversations – elements characterized as important to (or as interfering with) their goals – and how these structures enable or constrain learner agency. In the remainder of this section, I discuss each of these five sources of structure: *teacher*, *student*, *computer*, *classroom*, *school*.

Table 6.2 Sources of structure available and experienced in K-12 classrooms.

Source	Structure	Enable(s)	Constrain(s)
Teacher	Attitude	When teachers' curiosity, enthusiasm, and courage support experimenting with open-endedness	When teachers' fear and anxiety lead to controlling behavior
	Experience	When technical, content, and pedagogical knowledge align with aspirations	When too little expertise leads to overly-constrained activities; When too much expertise leads to solving problems for (rather than with) students
Student	Attitude	When students' curiosity, enthusiasm, and courage support experimenting with open-endedness; When student self-concept aligns with "problem solver" and/or "computational creator"	When students' reluctance, avoidance, and refusal discourage experimenting with new ways of learning
	Current Abilities	When opportunities are available to have students' abilities and expertise valued	When lack of abilities leads to too much challenge; When abundance of abilities leads to too little challenge
Computer	Authoring Environment	By providing learning opportunities different from other school-based technology experiences	When lacking particular features for certain domains (specifically, computer science)
	Online Community	By providing motivation of authentic audience	When evoking concerns about privacy, safety, and protection
	Educator Online Community	By serving as a source of agency-enabling ideas through stories, resources, and discussions	When serving as a source of agency-constraining ideas through stories, resources, and discussions
Classroom	Group	By providing a diversity of perspectives; When encouraging exchanges between group members	When group size and multiplicity of (potentially conflicting) needs lead to controlling behavior

Table 6.2 continued.

Source	Structure	Enable(s)	Constrain(s)
	Roles	When engaging in more collegial, mutually-supportive roles	When enacting "traditional" roles of teacher-as-expert, student-as-follower
	Activities	By providing opportunities for designing, personalizing, sharing, and reflecting	When neglecting some aspect of designing, personalizing, sharing, or reflecting; When favoring structure over agency
School	Intentions	By fostering a dedicated community supporting agency of both students and teachers	When time-regulated nature of school conflicts with needs of problem-solving, authentic challenges, and creativity
	Assessment	When supporting learner progress	When favoring simple assessment over meaningful assessment

Teacher: Attitude And Experience

In November 2008, I was in Copenhagen, Denmark attending a conference. Around the city, I encountered a series of large signs (e.g., Figure 6.2) that encouraged the viewer to "replace fear of the unknown with curiosity." I was reminded of these signs (which were promoting an art exhibition) and their message as I thought about how teachers described structures that can enable or constrain student agency.

As noted in Chapter 5, many teachers are afraid or anxious about Scratch, particularly when thinking about introducing Scratch in a way that supports open-ended design, an approach that encourages kids to define and develop solutions to their own problems. We know that teacher attitudes, feelings, and beliefs impact student learning – in general and when including technology and computation in teaching practice (Buckingham, 2007; Cuban, 2001; Schofield, 1995) – and Chapter 5 shared stories of how teacher anxiety can diminish the agency of young people in classrooms.

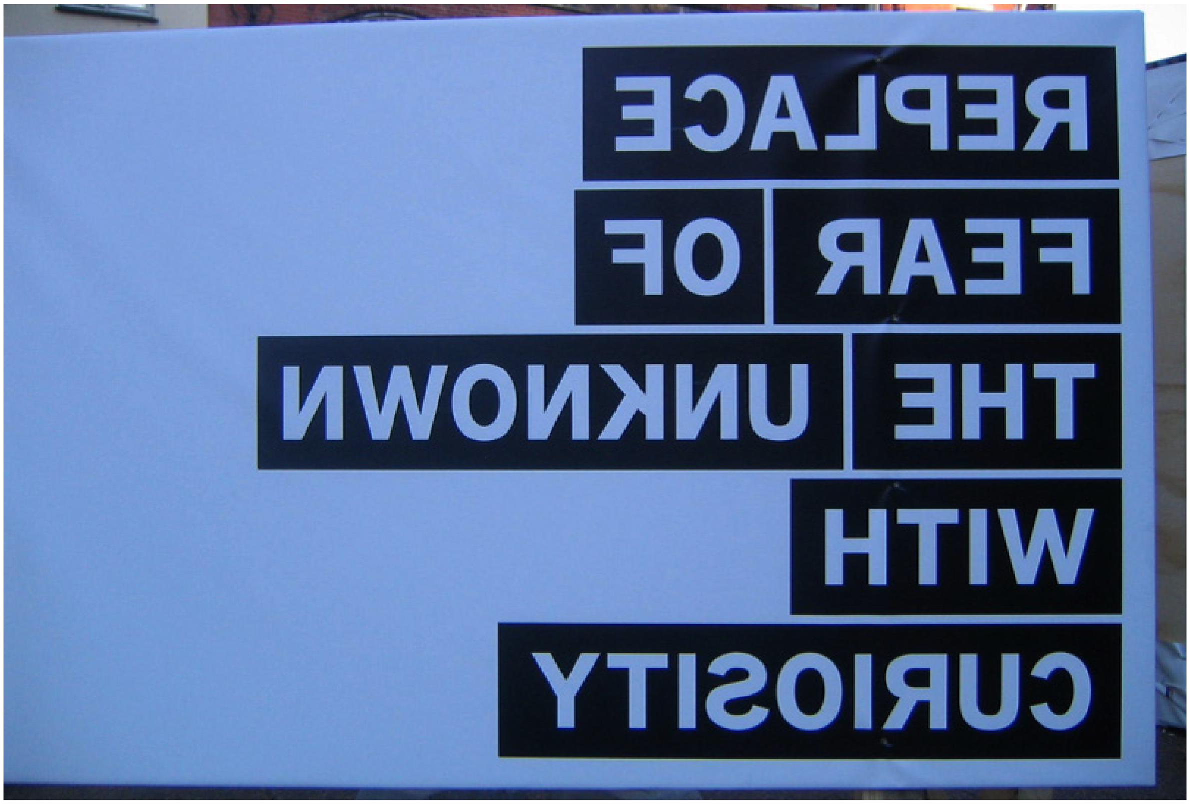


Figure 6.2
One of several large signs spotted around Copenhagen in November 2008.

While there are many external structures (as will be discussed later in this section) that explicitly or implicitly discourage teachers from trading *fear* for *curiosity*, several teachers noted a more personal barrier to change – altering one's teaching practice might imply that, previously, one had not been teaching effectively. Crawford reflected on his own experiences with change.

> Why is there change? You're dissatisfied with the way it's going. If you're a teacher, that's sometimes the case, but usually you're not because – this is another pet theory. If you say, "Oh, you're right. I should do more open-ended problem-solving, logical thinking, that's how kids are going to learn," then you're basically admitting to yourself, "Oh, the way I was doing it was crap and now I've failed these kids." That's the extreme case and I see that in myself even. And I feel guilty, even though I'm conscious of it. I'm like, "Oh, so I used to teach it that way, and now I'm teaching it this way." So I'm basically saying, "I was a horrible teacher and I

failed those kids." I think that's why there's some resistance to change.

Crawford, Middle School Teacher

Jackie shared a similar reflection regarding her previous approach to computing with her elementary school students.

> So now that I am done with teaching Scratch this term, I'm kind of looking back and saying, "OK. Basically I have been teaching secretarial skills to these kids." You know what I mean? I thought, "They have to learn how to do word processing. That's one of the things they need to learn." Now, I'm thinking, like, "Oh, that's a secretarial skill." So, I like that every Scratch lesson has some kind of thinking skill or computational thinking skill that we were trying to get them to wrap their heads around. I kind of felt like this Scratch experience was a lot more purposeful than a lot of the stuff that was going on in my classrooms before.
>
> Jackie, Elementary School Teacher

Teacher comfort and confidence was often connected to teacher experience and knowledge – their technological experience/knowledge (familiarity with Scratch), their content knowledge (familiarity with solving problems, programming concepts and practices), and their pedagogical knowledge (familiarity with teaching). Too little experience or knowledge in any one of these areas negatively impacted teacher comfort and confidence, leading them to adopt more rigid structures for their students. Too much experience or knowledge similarly undermined student agency, by not giving students space to explore, experiment, and build capacity as self-managing problem solvers. As described in Chapter 5, teaching Scratch using a constructionist approach does not necessitate expertise or mastery of Scratch, but rather an openness, curiosity, and vulnerability as a learner.

Examples

In *Negotiating Open-Endedness* (Ch. 5), Arnold's experience with his music teacher, Linda's lack of comfort leading to reduced student agency.

In *Legitimizing Learning* (Ch. 5), ScratchEd posts about online fears, Crawford's anxiety in response to implicit pressures to justify learning.

In *Building Culture* (Ch. 5), Clive's sympathy for student experience from trying it himself, Tara's feeling that one cannot know everything, Kent's encouragement for teachers to be curious learners.

Student: Attitude And Current Abilities

Like teachers, the attitudes and experiences (or current abilities) of students are important structures to consider when thinking about learning in K-12 classrooms. Kids bring a set of beliefs about themselves, about their teachers, about their peers, and about school environments – all of which impact the types of experiences that are possible in classrooms.

Examples
In *Negotiating Open-Endedness* (Ch. 5), Tara and Valerie's experiences with students who ask to be told what to do and how to do it.

As described in Chapter 5, students' negative attitudes or unwillingness to participate in self-directed learning created challenging conditions for teachers to support learner agency. Teachers described kids who, never having had opportunities in school to take responsibility for their learning in a way that connected to their interests and passions, were reluctant to take on this foreign approach to learning, given that it seemed more demanding with no apparent "incentive".

On the other end of the spectrum, positive student attitude and self-concept, as well as a willingness to explore, experiment, persevere, and be courageous, all serve as enabling structures in the classroom, just as they do for kids working at home. Teachers described how their work often involved cultivating and supporting the development of positive self-concept, working with students to think about themselves as being (or capable of becoming) problem-solvers, and finding ways to connect with students' motivations and interests.

Examples
In *Building Culture* (Ch. 5), Larissa's exploration of crocodile noises, Beau's case for games, Linda studying student code, Clare starting class sessions with student demos.

In addition to supporting young learners by encouraging them to pursue their passions and interests or make culturally-relevant connections (as we saw in Chapter 5), creating opportunities for students to share their abilities and expertise with each other is an important enabling factor. Students experience pride when their learning is valued by the teacher and, more importantly, valued by their peers.

Computer: Authoring Environment And The Online Communities

Examples
In Table 3.2, teachers' pathways into Scratch.

The teachers I spoke with were enthusiastic about the Scratch authoring environment, as they felt it represented a different type of engagement with technology than was otherwise available for students in school. Several of the teachers were familiar with (and, in some cases, even had extensive experience with) the Logo programming language, which they saw in a similar way. Giving students opportunities to be the designers of computational artifacts, rather than the consumers of already-designed computational artifacts, was seen as affording different types of learning opportunities than using a web browser, writing an essay with a word processor, or making a video with video editing software.

Examples
In *Supporting Problem-Solvers* (Ch. 5), Sabine's observation about the centrality of technology in the world, Clayton helping kids learn how to talk with programmers.

Just as it does for kids working on projects at home, the Scratch authoring environment makes it easy for students in classrooms to get started, while having sufficient flexibility to enable the creation of more complex projects as students progress. However, students in classrooms also face some of the same challenges as kids at home with respect to open-endedness and authoring-environment limitations. Computer science high school teachers, for example, discussed how the authoring environment lacked specific programming features necessary to satisfy computer science curricular requirements (e.g., Scratch's lack of named procedures). This constraint led some teachers to supplement their activities by using other programming languages.

Unlike the authoring environment, and despite its tremendous potential for enabling student agency, the online community is not as easy to incorporate into school activities as it is into home activities. For example, during a show-and-tell session at a Scratch educator meetup hosted at MIT, a middle-school teacher talked about her motivation for connecting kids with the online community.

Examples
In *Building Culture* (Ch. 5), "creating opportunities to share" strategy recommended by teachers, extending beyond the classroom walls.

> Coming from an English background, it was really important to me that there was a "publication" step. As a writer, the last step is always sharing, getting feedback, and critiquing. That's why I really bought into the whole idea of the website. My middle-schoolers are mostly inspired by the feedback they get from their peers and the gratification they get from sharing their projects in such a public way.
>
> Middle-school teacher

Examples
In *Legitimizing Learning* (Ch. 5), ScratchEd posts about online safety, high-school student's "Access denied" Scratch project.

Examples
In *Building Culture* (Ch. 5), "trying it yourself" strategy recommended by teachers, which involves accessing resources to support their learning.

But concerns about online access in the school setting, including privacy, protection, and plagiarism (as discussed in greater detail in Chapter 5) often constrained the possibilities.

Another online resource that was referenced frequently by teachers is the ScratchEd online community. Although not designed for students, teachers talked about how the structure of the online community facilitated learner agency and constructionist experimentation by enabling educators to connect with one another and share their experiences. There have been numerous examples within the ScratchEd community of how access to the experiences of others produced a shift in favor of learner agency. For example, a parent who was invited to speak to his son's 7th grade class about his career as computer programmer thought that it would be great for the kids in the class to have a hands-on experience with programming, and arranged with the teacher for Scratch to be installed. As the presentation approached, he considered different strategies for introducing the kids to Scratch. He posted to the ScratchEd forums in search of advice.

> I would like to give the kids the basic idea of a step-by-step algorithm with loops. But, I don't want to make it so complicated that the kids get bogged down or bored. I'm thinking about leading the kids through Scratch Getting Started. I'd be very grateful for any suggestions on good activities for this presentation, comments on what to expect, warnings of problems to be prepared for, etc.
>
> ScratchEd forum post

I responded, and suggested that he might be interested in another thread, *How do you introduce Scratch to beginners?* A few weeks later, the parent posted again, describing the success he had enjoyed with the students.

> Thanks for the links. The presentation went well and the kids loved Scratch. My original idea was to walk the kids through the Scratch Getting Started guide. After reading the thread, "How do you introduce Scratch to beginners?" I changed my

> approach. Instead of a step-by-step walk through, I decided to demonstrate Scratch quickly and then get out of their way. … The most common question was, "Where is the web site where I can download Scratch." I'm glad that I read the thread and changed my approach. Thanks a bunch.
>
> Follow-up to original ScratchEd forum post

The ScratchEd online community also provides teachers with access to an array of resources – resources that are developed by people at MIT and by members of the community, and have varying degrees of alignment with a constructionist approach to learning. For example, some lesson plans are highly didactic. But, even for resources that are intentionally designed with constructionist ideals in mind, designer intentions do not always align with how the resources are employed. For example, the Scratch curriculum guide that was developed by the ScratchEd team and released in 2011 was intended to be remixed and adapted by educators. Some educators, however, felt uncomfortable with making modifications, and opted to follow it extremely faithfully, even if there was a lack of clarity about the purpose for including particular activities.

> My students had never heard of the six-word stories, so that was interesting. And I apparently forgot what they were. But I talked to the English instructor at our school and she just thought it was such a cool idea, and I did too. So we were looking them up on the Internet and there were just some really funny ones up there and they really enjoyed that. Just because it was something different, you know? But I didn't quite see where that was going. I mean, it was cool that we had six-word stories – but then I'm like, "Um, I don't get it."
>
> Kirby, High School Business Education Teacher

Other educators felt comfortable with adapting the curriculum, but implemented changes that were (as discussed earlier) incongruent with the guide's aspirations. For example, one educator emphasized the value of optimized code, which I feel is not a significant consideration for those just learning how to program.

Classroom: Group, Roles, And Activities

Examples
In *Building Culture* (Ch. 5), Taylor creating a space where "this is what we do as learners".

In Chapter 5, I described teachers' main strategy for negotiating the complexity associated with an open-ended, design-based approach to learning: building a new culture of learning in the classroom. This necessitated thinking about the structure of the classroom in different ways – reimagining the nature of learning with/in a group, individuals' roles, and the learning activities.

Examples
In *Negotiating Open-Endedness* (Ch. 5), Clive managing many distractions, Clare's student helper's exasperation with impatient students, Linda's fears of too many directions.

Learning within a group, as happens in classroom settings, presents many challenges. For example, the size and diversity of the group can be intimidating, particularly when thinking about encouraging learners to pursue their individual interests. But size and diversity can be an asset in the structure of the classroom, presenting opportunities for teachers to encourage peer learning and mentorship, in turn providing greater support for learners to pursue their goals and develop as computational creators. Teachers described the satisfaction students experienced in learning from each other, as Larissa shared –

> I've always had students teaching because I found out that students communicate better with other students.
>
> Larissa, K-12 Teacher and Teacher Educator

They also described how quickly this peer learning spread throughout the classroom community, as Clayton shared –

> It's like how ten kids get some video game and then one of them learns where the key to the castle is. In 30 minutes, everyone knows! It's that kind of thing – they can really teach each other those kinds of things quite well.
>
> Clayton, Elementary School Teacher

This type of peer learning often necessitates new ideas about classroom roles and activities, dislodging some of the historical asymmetries of classroom-based learning environments. The teacher (as described in Chapter 5) does not serve as the source of all knowledge, and enters into authentic, engaged learning with students, helping students identify and pursue their learning goals, and sharing responsibility for the classroom culture. The students take on new responsibilities, for their own learning and for the learning of their peers in class.

These new roles are supported and reinforced through careful activity design, balancing the structure needed for capacity-building and sufficient scaffolding, while maintaining the open-endedness of engaging problem-solving, connecting to personal interests, and fostering creativity.

Examples
In *Building Culture* (Ch. 5), creating opportunities for learners to connect with each other (Sun encouraging community) and to think about roles in new ways (teacher as coach, with Linda and the jellyfish).

School: Intentions And Assessment

The teachers that I spoke with are passionate, dedicated educators, committed to designing learning environments that engage and enrich student experiences from day-to-day, and that will also meaningfully support and prepare students over the longer term, for full participation in 21st century society. Teachers' desires to foster students' capacities for problem-solving and self-expression are not primarily motivated by short-term, school-based performance concerns; rather they stem from beliefs about the fundamental project of schools. Larissa described her focus on supporting students as independent learners, thinkers, and decision-makers.

Examples
In *Supporting Problem-Solvers* (Ch. 5), ScratchEd post about fostering problem-solving capacities, Arnold fostering a love of problem solving, Taylor's aspiration for building confidence through creation and design.

> Students know they can come to me [for help], but they know that they have to take their own decisions. As a person in education, you might understand that one of the most important things we have to do with all these human beings is help them learn how to make their own decisions.
>
> Larissa, K-12 Teacher and Teacher Educator

But teacher intentions are often influenced by short-term, school-based performance concerns – the expectations of the school structure – that impinge and constrain learner agency in a variety of ways. Time was often cited by teachers, and by the kids I interviewed, as a constraint of school. For example, when I asked Jan about whether he would be interested in introducing Scratch into his school, he struggled to imagine a successful outcome, given the way in which school is structured into time blocks.

Examples
In *Negotiating Open-Endedness* (Ch. 5), challenges of time, number of students, and variability of student expertise and interests.

> There's always this conflict of schools. They are organized into one-hour time slots – and for creative projects, it's never enough. If you want to engage in creativity, it takes its time.
>
> Jan, 16 years old

Examples
In *Legitimizing Learning* (Ch. 5), Kirby's desires for something measurable, Jody's efforts toward assessing effort, Crawford's sense of justifying.

Other school expectations also served as constraints. As discussed in Chapter 5, expectations about demonstrating and providing evidence for student learning are high, yet methods are lacking for assessing and evaluating the types of learning aligned with teachers' intentions. Often, the aspects of learning that are easiest to assess are not the aspects most significant to students' development as learners with agency.

Chapter 7

THE BEST OF BOTH WORLDS

In this chapter, I argue that designers of agency-supporting learning environments, rather than setting structure in opposition to agency, should judiciously employ structure in order to amplify agency. I offer five strategies for designers of learning environments, suggesting opportune ways of introducing structure in the service of learner agency.

In the opening lines of *Experience and Education* (1938), John Dewey calls attention to the human tendency to dichotomize.

> Mankind likes to think in terms of extreme opposites. It is given to formulating its beliefs in terms of *Either-Or*, between which it recognizes no intermediate possibilities. (p. 1)

In this thesis, I have been preoccupied with the often-assumed dichotomy of *agency* (defined earlier as *a learner's ability to define and pursue learning goals*) versus *structure* (defined earlier as *rules, roles, and resources, both explicit and assumed*) in learning environments.

Chapter 1 outlined my theoretical framework for agency and structure, which argues for thinking about agency and structure as mutually constitutive, drawing on Giddens' structuration theory. Chapter 2 introduced some of the assumptions (and myths) about agency and structure that are often associated with the adoption of constructionism (and its high-agency aspirations) as an epistemological stance. Chapter 3 described my methodological approach to studying agency and structure, which emphasized interview and observation. Chapters 4, 5, and 6 examined the tensions between structure and agency in two settings for computational creation – kids working with Scratch at home through the online community, and teachers working with Scratch in K-12 classrooms. These chapters described the learning processes in each setting (Chapter 4 and Chapter 5) and the key structures that enable or constrain learner agency (Chapter 6).

Here, I connect the threads from the previous chapters, offering strategies for thinking beyond an *either-or* approach to agency/structure, in favor of exploring *intermediate possibilities*. I argue that designers of agency-supporting learning environments, rather than setting structure in opposition to agency, should judiciously employ structure in order to amplify agency.

This chapter is organized into two parts. The first part – *Either-Or* – illustrates, with a story, how too much and/or too little structure can inhibit learner agency. The second part – *Intermediate Possibilities* – provides advice for designers of learning environments, suggesting opportune ways of introducing structure in the service of learner agency.

EITHER-OR

During my time as a graduate student at MIT and as a member of the Scratch Team, I have had many opportunities to help people learn about Scratch. In workshops and presentations, I usually begin by describing Scratch and why it is important. I demonstrate the features of the authoring environment and online community, and talk about participation from a quantitative perspective, to give a sense of scale. I share what I think is most interesting about the online community as a space for learning – the incredible diversity of projects that kids create and share, and the types of connections that kids establish with each other, from giving feedback in comments, to remixing each other's work, to making projects together.

After talking about the online community as a learning environment, I explain my interest in supporting these types of activities in schools. I argue that not all young people have opportunities for these types of learning experiences at home, and that including Scratch in K-12 classrooms is one strategy for broadening participation in computational culture.

Occasionally, however, this strategy is questioned – "Isn't it OK for them to be separate? How can the type of learning happening at home even happen in the classroom? Won't schools take away everything that you think is great about the online community?"

Certainly, there are occasions in which Scratch's translation from out-of-school to in-school is unsuccessful. For instance, several years ago, I was asked to introduce 30 high school girls to the work of our research group, and to provide a hands-on experience with Scratch. As the participants arrived for the workshop, I noticed one girl scowling, with arms crossed – a markedly different comportment than her more animated companions. My colleagues and I welcomed the girls to our lab, and described our plan for the afternoon's workshop: a brief overview of the Media Lab, an introduction to Scratch, and then time for creating their own Scratch projects. The girl's scowl deepened.

We showed the girls some projects that others their age had created and shared in the online community, and then introduced the theme of the afternoon's activity – an interactive dance party. We quickly modeled how to start a dance party project by adding music, a background, and several dancers, and then paired the girls to develop their own dance party projects. The girls worked for an hour with their partners, consulting their neighbors, importing their favorite songs, and taking photos of themselves to put into their projects. We took time at the end for the girls to share their projects with each other, which provoked laughter, questions, and applause.

In all of the activity, I lost track of the scowling girl, until she approached me and said, "I hate Scratch." It was the first time I had experienced such a negative reaction to Scratch. I waited for her to say something else.

She continued, amending her statement, "I *used* to hate Scratch." She explained that she had been dreading the afternoon's session because of a bad experience with Scratch at school. Her teacher had assigned the students to draw geometric shapes with Scratch's pen tool and had required them to follow the teacher's actions step-by-step. They had spent a week with Scratch in this way, building up increasingly complex programs, but with no opportunities to customize or explore, and discouraged from interacting with each other. In that setting, the girl didn't understand Scratch. It didn't mean anything to her. Why would anyone want to do anything like this? Why was it interesting or important to her – or to anyone else? Her inability to distinguish pedagogy from technology had led her to expect the worst from the afternoon workshop, but she was surprised by how much she (and the other girls in her group) had enjoyed Scratch. She asked for a Scratch postcard, so that she would know how to download Scratch at home.

As this story illustrates, imposing too much structure can constrain learner agency. Designing was constrained to following the teacher's process. Personalizing was discouraged and the subject matter was not personally appealing. Sharing was prohibited, with individual students working at individual computers on individual projects. Reflecting on the design activities was not promoted, leaving the girl to question why the activities were being done at all.

Given these kinds of experiences, it isn't hard to understand the motivation behind questions like, "Won't schools take away everything that you think is great about the online community?" These questions reflect underlying assumptions that learning at home is low structure and high agency, that learning at school is high structure and low agency, and that agency and structure – and out-of-school and in-school settings – are fundamentally incompatible.

But although excessive structure undermines the agency of young people, I argue that the response should not be to dismiss structure more generally. As described in previous chapters, a lack of structure does not equal agency.

At the other end of the spectrum, Chapter 4 illustrates the ways in which too little structure can constrain learner agency. Kids who go to the online community, create an account, but don't understand the range of creative possibilities afforded by Scratch. Kids who have big visions for a Scratch project, but can't figure out what they need to learn to actualize their visions. Kids who want to find another Scratcher to work with, but the size of the community makes it difficult for them to be seen or to find someone with shared interests.

Ostensibly, kids who learn primarily in the online community have a higher degree of learner agency – a freedom to choose what and how to learn – than kids who learn primarily in classrooms. But in addition to confusing a lack of structure with agency, this mistakenly assumes that all types of freedom are equally desirable. I began this chapter with the first lines of John Dewey's *Experience and Education* – and I return to Dewey for a critical reflection on learner freedom.

> The ideal aim of education is creation of power of self-control. But the mere removal of external control is no guarantee for the production of self-control. It is easy to jump out of the frying-pan into the fire. It is easy, in other words, to escape one form of external control only to find oneself in another and more dangerous form of external control. Impulses and desires that are not ordered by intelligence are under the control of accidental circumstances. It may be a loss rather than a gain to escape from the control of another person only to find one's conduct dictated by immediate whim and caprice; that is, at the mercy of impulses into whose formation intelligent judgment has not entered. A person whose conduct is controlled in this way has at most only the illusion of freedom. Actually he is directed by forces over which he has no command. (p. 75)

There is enormous potential for schools to help support the development of this type of self-control, a form of structure that is the basis of learner agency, enabling young people to flourish as learners with the ability to define and pursue goals. Teachers need greater support to reconceptualize their roles in a way that enables these changes – including emotional support, classroom support, and institutional support.

And just as we should begin to think about agency and structure not as separate, but as mutually-reinforcing concepts, we have a similar opportunity to reconsider the dichotomy between out-of-school and in-school settings. We need to think about learning more continuously, the ways that out-of-school and in-school activities can support each other, and the ways in which kids can carry learning across settings, beyond facile ideas of "transfer". Learning is a lifelong (across ages, not just K-12 and college) and life-wide (across settings, not just school) enterprise, with the life-wideness accelerated by network and other digital technologies (Banks et al., 2007; Dede, 2011; DML Research Hub, 2012; Pellegrino & Hilton, 2012; Roschelle, Bakia, Toyama, & Patton, 2011).

There are intermediate possibilities to be explored with *both* the agency/structure relationship and the out-of-school/in-school relationship – it is not *either-or* with agency and structure, and need not be *either-or* with out-of-school and in-school settings. We can have the best of both worlds. But how do we design for the best of both worlds, seeking the intermediate possibilities?

INTERMEDIATE POSSIBILITIES

Agency-centered learning that takes place within/across a variety of settings is not merely an aspiration. Chapters 4, 5, and 6 presented glimpses of the possibilities – and I end this chapter with five sets of concrete strategies and recommendations for designers of learning environments who aim to support the development of learner agency. The strategies – *introduce possibilities*, *encourage experimentation*, *support access to resources*, *cultivate connections with others*, and *create opportunities for reflection* – suggest ways to productively introduce structure into learning environments.

These strategies represent a synthesis of commonalities in the experiences of kids working with Scratch at home in the Scratch online community (from Chapter 4) and the experiences of teachers working with Scratch in K-12 classrooms (from Chapter 5), filtered through my own experiences as a designer of agency-centric learning environments focused on computational creation. For each strategy, I provide examples of how different types of learning-environment designers – *teachers* in K-12 classrooms, *parents* at home, and *developers* of learning technologies (using the Scratch Team as a concrete example) – have employed structure to support agency in computational creation. Although the examples stem from the context of computational creation with Scratch, these strategies are not, in general, limited to a particular domain or particular tool.

Strategy 1: Introduce The Possibilities

When encountering a new area of learning, the entry-points and trajectories of participation and learning are often not self-evident. Learners might wonder: *What can I do? Why might I want to do it?*

There are opportunities within learning environments for structure to bring clarity to these trajectories by introducing and describing the possibilities of what the learner can do, both broadly and in a focused manner at particular waypoints. In the broad case, this might involve explaining what a particular tool makes possible, discussing how an activity is connected to a learner's personal interests, or how an activity has larger social or technical relevance. In the more focused case, this might involve explaining what is possible for a learner at a particular moment, connecting to current abilities – like presenting an appropriately scoped and scaled learning challenge.

Examples

- Teachers: A middle-school teacher was facilitating a digital media production course that included a unit on interactive media creation with Scratch. Knowing how passionate her students are about music, she searched the Scratch online community for animated music video projects. She began the initial class by dimming the lights, pronouncing, "Look at what you'll be able to create in just a few weeks," and screening several of the music video projects for her students.
- Parents: A parent read a news article about Scratch and thought that his daughter might be interested in trying it. He searched for a tutorial and found an overview video that explained what Scratch is and illustrated what you can do with it. He shared the video with his daughter.
- Developers: The Scratch Team decided to include a series of Scratch projects as part of the distribution of the authoring environment. The sample projects span a range of genres – animation, games, art, simulation, and more – and include projects of varying complexity within each genre.

Strategy 2: Encourage Experimentation

Learning through designing involves negotiating constraints – it requires a learner to adopt a stance of flexibility and experimentation in relation to their learning goals. Learners might wonder: *What are the goals I want to pursue? What are different pathways to achieving my goals?*

Opportunities exist within learning environments to use structure in support of experimentation. Designers can support learners in experimenting with both the *what* of their learning (e.g., encouraging diversity in creation) and the *how* of their learning (e.g., highlighting different pathways or strategies for achieving goals). A learner's willingness to vary subject and method is, however, in tension with a simultaneous desire to persist with particular experiments. The balance between breadth (experimenting) and depth (persisting) should be supported, along both cognitive and affective dimensions.

Examples

- Teachers: An elementary teacher encouraged her third-grade students to define a project to work on. But one of her students, who was developing a playful story about an Arctic mishap, struggled with programming the visual behaviors of the story. The teacher helped the student identify several strategies for debugging the project. When the student shared her project with others, she described the array of troubleshooting strategies they employed, including her approach to experimenting with blocks in the *looks* category, and described how important it was to have "grit" while experimenting.
- Parents: An 8-year-old Scratcher was frustrated by his project, which wasn't working, and went to his mother, teary-eyed, for advice. His mother gave him a hug, and recommended working on a different project until he had an idea for how to fix the problem with his original project. Feeling better, he returned to his room and made a list of potential ideas for new projects.
- Developers: The Scratch Team designed the Scratch authoring environment to be "tinkerable" – enabling something large to be built up by playing with smaller pieces, rather than breaking apart a larger plan into smaller components (as is often the case with other tools for computational creation). In Scratch, this enables young designers to experiment with different blocks and observe their behavior and impact on a project in real-time.

Strategy 3: Support Access To Resources

In agency-centered learning environments, learners take ownership of and responsibility for learning goals, instead of primarily following the ambitions and direction of others. But in order to achieve their goals, learners require access to resources to support the pursuit of the individualized pathways. Learners might wonder: *What can help me achieve my goals?*

Ideally, designers will make resources available that are appropriately-timed and appropriately-accessible (both in format and complexity) for the learner. Resources may be centralized (e.g., a resource accessed by all members of the learning environment simultaneously) or decentralized (e.g., just-in-time resources accessed on demand by learners). Resources may take on a variety of formats (e.g., text, video, audio) and complexity (e.g., resources for beginners, resources for learners who are further along).

Examples

- Teachers: A high-school animation teacher made the *Scratch cards* (a resource developed by the Scratch Team) available to her students. The cards each describe something that one might want to do with Scratch, such as creating a character that dances back and forth or using variables to maintain a score in a project. The teacher used the Scratch cards as a model, encouraging her students to develop new resources based on the challenges they experienced when making Scratch projects.
- Parents: A mother saw her daughter wanting to make more sophisticated games with Scratch, but unable to make progress on her goal. She bought her daughter a book that included a series of tutorials for creating games.
- Developers: On the main page of the Scratch online community, the Scratch Team included links to resources designed to help people get started with Scratch. One link points to "Scratch Tours", which are collections of thematically-related, annotated projects, while another link points to a collection of introductory video tutorials. A link is provided to the ScratchEd online community, which contains

numerous resources for teachers (and is also available to young learners and parents).

Strategy 4: Cultivate Connections With Others

Learning is not an individual process – learners can benefit from being connected with others. These connections can take different forms, with others potentially serving key roles as advisors (e.g., providing advice for challenges), as collaborators (e.g., jointly pursuing a learning goal), as audience (e.g., showing appreciation for creative work), and/or as advisees (e.g., someone with whom to share one's understanding). Learners might wonder: *Who can help me achieve my goals? Who might work with me to define and purse goals? Who might appreciate my achievements? Who might I help achieve their goals?*

Cultivating connections between learners and others involves at least two components: (1) helping learners identify potential connections (i.e., matchmaking), and (2) supporting positive interactions within those connections (i.e., respectful, productive, and mutually beneficial). Designers can introduce structures that support connection-making processes (e.g., introducing learners to those who have compatible and complementary interests, or grouping learners with those who have divergent interests as a way to broaden learners' perspectives). Further, designers can introduce structures that contribute to the success of these connections (e.g., by providing partnered learners with ideas for how to give each other constructive criticism).

Examples

- Teachers: At the end of every class, an upper-elementary school teacher encouraged his students to post challenges that they experienced with their Scratch projects on a whiteboard. At the beginning of each subsequent class, he asked students to select someone else's challenge and then meet for 10 minutes to discuss possible solutions. At the end of the Scratch unit, the teacher invited a lower-elementary class and several of his colleagues to attend final class project demonstrations. His students introduced the younger kids, their teachers, and the school principal to Scratch.

- Parents: A father introduced his son to Scratch and they worked on making projects together. But the son quickly outpaced the father's abilities – and the father wished he could connect his son to another young Scratcher. He learned about a Scratch event being organized by a local university and brought his son. At the event, his son met another young Scratcher and they enthusiastically worked on a project together at the event. They continued working on the project through the online community and occasional face-to-face meetings.
- Developers: When the Scratch online community initially launched, the Scratch Team included an online forum where members could ask and answer questions, but the team was uncertain about the extent to which young Scratchers would use this feature. Much to the team's surprise, some kids in the community use the forums extensively – giving and receiving help, establishing partnerships, and advertising creations.

Strategy 5: Create Opportunities For Reflection

Learners need opportunities to reflect on their experiences – and the "learners might wonder" questions from the previous four strategies are examples of reflective prompts. Reflection happens both in real-time and in retrospect (e.g., Schön's reflection-in-action vs. reflection-on-action, as described earlier), and involves both cognitive and affective dimensions of learning experiences.

The designer of a learning environment can make use of structure to create opportunities for learners to articulate and negotiate these types of reflective prompts. Structures can be employed to support learners in identifying and taking steps to resolve the gap between what is currently known and what is not yet known. Structures can also be employed to support learners in negotiating feelings about their experiences, and the (expected) vulnerability of being a learner with responsibility for one's own learning. Learning can be simultaneously hard and fun (as described by Papert and echoed by kids in Chapter 4) – and structures that support reflection can support awareness of when learning might be moving from eustress to distress.

Examples

- Teachers: A high-school computer science teacher asked his students to maintain design journals accompanying their Scratch projects. As students came into class each day, there were questions on the front board to which the students would respond in their journals. The questions spanned a range of topics, and encouraged the students to reflect on their experiences. What is your plan for today? What did you do yesterday that you were most proud of? What are three things you are able to do now that you weren't able to do when we first started?
- Parents: A 12-year-old girl felt creatively blocked. She had created several Scratch projects, but didn't know what she should create next. She went to her mother for assistance. Although her mother didn't know much about Scratch, she was able to help her daughter by asking questions about the types of projects that she had already created and then asking her what advice she would give to someone who was similarly stuck.
- Developers: The Scratch Team experimented with different ways to explicitly encourage and elicit reflection and reflective response. One Master's thesis explored augmenting the Scratch authoring environment with reflective prompts (Rosenbaum, 2009). The next generation of Scratch includes an area on a designer's profile page for the designer to describe what they are currently working on, as well as a summary of what they have been doing.

The process of designing learning environments, particularly those that emphasize learner agency, is complex and multi-faceted, with numerous factors shifting and in tension with one another. The five strategies presented in this chapter are intended to serve as points of focus in this shifting landscape, but also as points of continuous negotiation and iteration. The pursuit of intermediate possibilities between agency and structure is, to borrow an aphorism from Rogers (1961), "a direction, not a destination."

Epilogue

LIFE AFTER SCRATCH

In this chapter, I reflect on the future from three perspectives – as a researcher, as a designer, and as a learner.

As I am writing this, members of the Scratch Team are actively developing the next generation of Scratch. Scratch 2.0 represents the most significant changes to the Scratch authoring environment and online community since Scratch 1.4 was released in 2009.

These changes make me think about the future. I wonder what new opportunities and new challenges await the kids and teachers who have so deeply invested in computational creation and computational culture.

I consider these opportunities and challenges from three perspectives – as a researcher, as a designer, and as a learner.

Researcher

As a researcher, I think about new questions opened up by the work described in this thesis.

First, this work has illustrated the complex system of structures that young computational creators encounter, rely on, negotiate, and build up around themselves – contrary to "digital native" narratives, it is not just kids "doing Scratch on their own". Where do these experiences fit on a pathway/trajectory of participation? What happens next? What do these experiences lead to?

Second, this work has focused on kids who have, in relation to the larger Scratch community, been very successful. Their experiences – their aspirations, the challenges they faced, the strategies they developed to overcome challenges, how they validated their experiences – were predominantly positive and, as such, Chapter 4 might be read as uncritical. I would argue that this is not due to a lack of criticality, but due to my desire to faithfully and respectfully represent the experiences of these particular kids. But what about the kids who have less positive experiences?

Some kids may find entry-points, but have negative interactions on the Scratch community site. I describe instances of negative interactions in Chapter 4 and Chapter 6, as well as in another paper (Brennan, 2011). But other kids may not find entry-points or ways of participating. What about the kids who are unable to (or choose not to) make sense or take advantage of the structures that they encounter? Is the lack of connection due to a lack of interest or a lack of self-directedness as a learner? How might a lack of connection be related to the increasing size of the online community – making it harder to find what is most valuable to an individual learner, or harder to focus, with many demands on learner attention?

Third, in an era where learning is increasingly unleashed from school settings, this work has illustrated the special role that teachers can play in supporting young people's development of agency. But it has also described the complexity of supporting learner agency in school settings, and the numerous barriers that make the promise of supporting agency in school seem distant. How can teachers be better supported in thinking about themselves as designers of learning environments and as advocates for learner agency? How can teachers be better supported in enacting and assessing constructionist approaches to learning within the current realities of schools? How do kids in classrooms experience agency in computational creation? How do these experiences migrate across settings and across activities, both computational and otherwise?

Designer

As a designer, I think about how the kids and teachers that I have worked with have been impacted by their experiences with Scratch. In particular, I think about how kids' experiences of designing with Scratch, no matter the setting, might support how they re/design themselves, their future lives, and the world around them.

Five years of work have shown me a variety of possibilities. Some engagements with computational creation are quite brief, providing an interesting childhood experience among thousands of interesting childhood experiences, with unknown impact and influence on future activities and interests. Some engagements are intense and extensive, yielding hundreds of projects, developed over hundreds of hours, and, sometimes, sparking interest in a career. And many engagements are somewhere in between, with computation seen as enjoyable and interesting, and perhaps applicable to a future career – as a veterinarian, a café owner, or an Olympic luge athlete (as I learned in my conversations with young Scratchers).

Whatever the nature and extent of the engagement, I aspire for young people to have opportunities to engage creatively with the computational culture in which they are immersed. This is not solely a matter of participating in the workforce of the 21st century or of being a contributing member of society. It is also fundamentally about the individual and how they see themselves as learners with agency, capable of great creativity, as "it is creative apperception more than anything else that makes the individual feel that life is worth living" (Winnicott, 1971, p. 87).

Learner

As a learner, I think about how I have benefited from and through the Scratch and ScratchEd communities, and how I have experienced the constructionist learning I hope for others. I have designed – creating activities for the Scratch website, planning Scratch Day events and conferences, building websites for teachers and event organizers, and making resources for teachers and their students. I have personalized – figuring out what I'm passionate about and contributing to the Scratch research base in my own way. I have shared – working with interns, members of Lifelong Kindergarten, teachers, and young Scratchers. And, of course, I have reflected – struggling with challenges, but always experimenting, working towards the next iteration, trying to make things better.

The last interview that I conducted for this work was with Lana, who was also the first Scratcher that I had the opportunity to talk with. As I prepared for the interview by reviewing the transcript from our first conversation in 2007 and my half-decade of field notes, I anticipated that there would be a feeling that so much had changed, that so much was different now. Both she, now 17, and the Scratch online community had grown up considerably.

But as our conversation began, I felt a familiarity and comfort, almost as though we were picking up a conversation that we had only recently set aside. I explained that I was interested in her experiences, particularly given her long-term participation in, and perspectives on, the community. She told me that there was so much to the story and so much to share that she wasn't sure where to begin.

I reassured her that she could start wherever she wanted. She paused, taking a moment to collect her thoughts, and then began, with a laugh. "When I think about my life, I almost divide it into two parts – life before Scratch and life after Scratch."

I couldn't help but smile. "Me, too."

References

Ackermann, E. (1996). Perspective-taking and object construction: Two keys to learning. In Y.B. Kafai & M. Resnick (Eds.), Constructionism in practice: Designing, thinking, and learning in a digital world (pp. 25-35). New Jersey: Lawrence Erlbaum.

Adamic, L.A., & Huberman, B.A. (2002). Zipf's law and the Internet. Glottometrics, 3, 143-150.

Anderson, T., & Shattuck, J. (2012). Design-based research: A decade of progress in educational research? Educational Researcher, 41(1), 16-25.

Anthony, W. (1973). Learning to discover rules by discovery. Journal of Educational Psychology, 64(3), 325-328.

Appadurai, A. (1996). Modernity at large: Cultural dimensions of globalization. Minneapolis, MN: The University of Minnesota Press.

Bandura, A. (1997). Self-efficacy: The exercise of control. New York, NY: W.H. Freeman.

Bandura, A. (2001). Social cognitive theory: An agentic perspective. Annual Review of Psychology, 52, 1-26.

Banks, J.A., Au, K.H., Ball, A.F., Bell, P., Gordon, E.W., Gutierrez, K.D., et al. (2007). Learning in and out of school in diverse environments: Life-long, life-wide, life-deep. Retrieved August 30, 2012, from http://life-slc.org/docs/Banks_etal-LIFE-Diversity-Report.pdf

Barab, S.A., Barnett, M., & Squire, K. (2002). Developing an empirical account of a community of practice: Characterizing the essential tensions. Journal of the Learning Sciences, 11(4), 489-542.

Barab, S.A., MaKinster, J.G., Scheckler, R. (2003). Designing system dualities: Characterizing a web-supported professional development community. The Information Society, 19, 237-256. doi:10.1080/01972240390210064

Begel, A. (1996). LogoBlocks: A graphical programming language for interacting with the world. Advanced undergraduate project report, Massachusetts Institute of Technology. Retrieved July 10, 2012, from http://research.microsoft.com/en-us/um/people/abegel/mit/begel-aup.html

boyd, d., & Crawford, K. (2012). Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon. Information, Communication & Society. doi:10.1080/1369118X.2012.678878

Bransford, J., Brown, A.L., Cocking, R.R., Donovan, M.S., & Pellegrino, J.W. (2000). How people learn: Brain, mind, experience and school (Rev. ed.). Washington: National Academies Press.

Brennan, K. (2011). Mind the gap: Differences between the aspirational and the actual in an online community of learners. CSCL'11 conference proceedings, 342-349.

Brennan, K., Resnick, M., & Monroy-Hernandez, A. (2010). Making projects, making friends: Online community as catalyst for interactive media creation. New Directions for Youth Development, 2010(128), 75-83.

Brennan, K., & Resnick, M. (2012). Imagining, creating, playing, sharing, reflecting: How online community supports young people as designers of interactive media. In N. Lavigne and C. Mouza (Eds.), Emerging Technologies for the Classroom: A Learning Sciences Perspective. doi:10.1007/978-1-4614-4696-5_17

Brennan, K., Valverde, A., Prempeh, J., Roque, R., & Chung, M. (2011). More than code: The significance of social interactions in young people's development as interactive media creators. In T. Bastiaens & M. Ebner (Eds.), Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2011 (pp. 2147-2156). Chesapeake, VA: AACE.

Brown, A.L. (1994). The advancement of learning. Educational Researcher, 23(8), 4-12.

Brown, A.L., & Campione, J.C. (1996). Psychological theory and the design of innovative learning environments: On procedures, principles and systems. In L. Schauble and R. Glaser (Eds.), Innovations in Learning: New Environments for Education (pp. 234-265). Hillsdale, NJ: Lawrence Erlbaum.

Brown, J.S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. Educational Researcher, 18(1), 32-42.

Bruckman, A. (1997). MOOSE Crossing: Construction, community, and learning in a networked virtual world for kids. Doctoral dissertation, Massachusetts Institute of Technology.

Bruckman, A. (1998). Community support for constructionist learning. Computer Supported Cooperative Work, 7, 47–86.

Bruckman, A. (2006). Learning in online communities. In K. Sawyer (Ed.), Cambridge Handbook of the Learning Sciences (pp. 461-472). New York, NY: Cambridge University Press.

Bruckman, A., & Resnick, M. (1995). The MediaMOO project: Constructionism and professional community. Convergence, 1(1), 94-109.

Brunvand, S., Fishman, B., & Marx, R. (2005). Moving professional development online: Meeting the needs and expectations of all teachers. In J.R. Dangel & E. Guyton (Eds.), Research on alternative and non-traditional education (pp. 205-232). Oxford, UK: Association of Teacher Educators.

Buckingham, D. (2007). Beyond technology: Children's learning in the age of digital culture. Malden, MA: Polity.

Buckingham, D., & Sefton-Green, J. (2003). Gotta catch 'em all: Structure, agency and pedagogy in children's media culture. Media, Culture, & Society. 25(3), 379-399.

Buckingham, D., & Willett, R. (Eds.). (2006). Digital generations: Children, young people, and new media. London: Routledge.

Burdick, A., & Willis, H. (2011). Digital learning, digital scholarship, and design thinking. Design Studies, 32(6), 546-556.

Carter, L. (2006). Why students with an apparent aptitude for computer science don't choose to major in computer science. Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education, Houston, Texas, USA.

Chopra, A. (2012). New summer jobs + commitments plan to introduce low-income youth to technology-related skills. Retrieved January 20, 2012 from http://www.whitehouse.gov/blog/2012/01/17/new-summer-jobs-commitments-plan-introduce-low-income-youth-technology-related-skill

Clandinin, D.J., & Connelly, F.M. (2000). Narrative inquiry: Experience and story in qualitative research. San Francisco: Jossey–Bass.

Cole, M., & Wertsch, J.V. (1996). Beyond the individual-social antinomy in discussions of Piaget and Vygotsky. Human-Development, 39(5), 250-256.

Collins, A. (2006). Cognitive apprenticeship. In K. Sawyer (Ed.), Cambridge Handbook of the Learning Sciences (pp. 47-60). Cambridge: Cambridge University Press.

Collins, A., & Halverson, R. (2009). Rethinking education in the age of technology: The digital revolution and schooling in America. New York, NY: Teachers College Press.

Craig, R. (1956). Directed versus independent discovery of established relations. Journal of Educational Psychology, 47(4), 223-234.

Csikszentmihalyi, M. (1990). Flow: The psychology of optimal experience. New York, NY: HarperCollins.

CSTA. (2006). A model curriculum for K-12 computer science: Final report of the ACM K-12 Task Force Curriculum Committee (2nd ed.). Retrieved January 15, 2012 from http://csta.acm.org/Curriculum/sub/CurrFiles/K-12ModelCurr2ndEd.pdf

Cuban, L. (2001). Oversold and underused: Computers in the classroom. Cambridge, MA: Harvard University Press.

Damşa, C.I., Kirschner, P.A., Andriessen, J.E.B., Erkens, G., & Sins, P.H.M. (2010). Shared epistemic agency: An empirical study of an emergent construct. Journal of the Learning Sciences, 19(2), 143-186.

de Certeau, M. (1984). The practice of everyday life. Berkeley: University of California Press.

Dede, C. (2011). Emerging technologies, ubiquitous learning, and educational transformation. Lecture Notes in Computer Science, 6964/2011, 1-8. doi:10.1007/978-3-642-23985-4_1

De Souza, C.S., & Preece, J. (2004). A framework for analyzing and understanding online communities. Interacting with Computers, 16(3), 579-610.

Dewey, J. (1938). Experience and education. New York, NY: Collier Books.

Dexter, S., Anderson, R., & Ronnkvist, A. (2002). Quality technology support: What is it? Who has it? And what difference does it make? Journal of Educational Computing Research, 26(3), 265–285.

Dexter, S., Seashore, K., & Anderson, R. (2002). Contributions of professional community to exemplary use of ICT. Journal of Computer Assisted Learning, 18(4), 489–497.

DML Research Hub. (2012). What is connected learning? Retrieved September 1, 2012, from http://connectedlearning.tv/infographic

Donath, J. (2007). Signals in social supernets. Journal of Computer-Mediated Communication, 13(1), article 12. Retrieved September 5, 2012, from http://jcmc.indiana.edu/vol13/issue1/donath.html

Duffy, T.M., & Cunningham, D.J. (1996). Constructivism: Implications for the design and delivery of instruction. In D.H. Jonassen (Ed.), Handbook of Research on Educational Communications and Technology (pp. 170-198). Mahwah, NJ: Lawrence Erlbaum.

Dweck, C. (2000). Self-theories: Their role in motivation, personality, and development. Lillington, NC: Psychology Press.

Emirbayer, M., & Mische, A. (1998). What is agency? The American Journal of Sociology, 103(4), 962-1023.

Farooq, U., Schank, P., Harris, A., Fusco, J., & Schlager, M. (2007). Sustaining a community computing infrastructure for online teacher professional development: A case study of designing Tapped In. Computer Supported Cooperative Work, 16(4-5), 397-429. doi:10.1007/s10606-007-9049-0

Feynman, R.P. (1965). New textbooks for the “new” mathematics. Engineering and Science, 28(6), 9-15.

Fischer, G. (2002). Beyond “couch potatoes”: From consumers to designers and active contributors. First Monday, 7(12).

Fischer, G., & Nakakoji, K. (1997). Computational environments supporting creativity in the context of lifelong learning and design. Journal of Knowledge-Based Systems, 10(4), 21-28.

Fishman, B.J., & Davis, E.A. (2006). Teacher learning research and the learning sciences. In K. Sawyer (Ed.), Cambridge Handbook of the Learning Sciences (pp. 535-550). Cambridge: Cambridge University Press.

Flavell, J.H. (1979). Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry. American Psychologist, 34(10), 906-911. doi:10.1037/0003-066X.34.10.906

Foster, A. (2005). Student interest in computer science plummets. The Chronicle of Higher Education, 51(38). Retrieved January 10, 2012, from http://chronicle.com/article/Student-Interest-in-Computer/10912/

Freeman, J. (1972). The tyranny of structurelessness. Berkeley Journal of Sociology, 17(1972-1973), 151-164.

Fuller, H. (2000). First teach their teachers: Technology support and computer use in academic subjects. Journal of Research on Computing in Education, 32(4), 511–537.

Gardner, H. (1983). Frames of mind: The theory of multiple intelligences. New York, NY: Basic Books.

Gardner, H. (1991). The unschooled mind: How children think and how schools should teach. New York, NY: Basic Books.

Gardner, H. (1999). Intelligence reframed: Multiple intelligences for the 21st century. New York, NY: Basic Books.

Giddens, A. (1984). The constitution of society: Outline of the theory of structuration. Berkeley, CA: University of California Press.

Gilligan, C. (1982). In a different voice: Psychological theory and women's development. Cambridge, MA: Harvard University Press.

Golder, S.A., & Donath, J. (2004). Social roles in electronic communities. Paper presented at the Association of Internet Researchers conference, Internet Research 5.0, Brighton, England.

Goldhaber, M.H. (1997). The attention economy and the Net. First Monday, 2(4).

Goodfellow, R. (2005). Virtuality and the shaping of educational communities. Education, Communication, and Information, 5(2), 113-129.

Guzdial, M., & Forte, A. (2005). Design process for a non-majors computing course. Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education, St. Louis, Missouri, USA, 361-365.

Harel, I., & Papert, S. (1990). Software design as a learning environment. Interactive Learning Environments, 1(1), 1-32.

Hargittai, E. (2009). An update on survey measures of web-oriented digital literacy. Social Science Computer Review, 27(1), 130-137.

Hayles, N.K. (2005). My mother was a computer: Digital subjects and literary texts. Chicago: University of Chicago Press.

Hennessey, B.A., & Amabile, T.M. (1998). Reality, intrinsic motivation, and creativity. American Psychologist, 53(6), 674–675. doi:10.1037/0003-066X.53.6.674

Hew, K.F., & Hara, N. (2007). Empirical study of motivators and barriers of teacher online knowledge sharing. Educational Technology Research and Development, 55(6), 573-595.

Hine, C. (2008). Virtual ethnography: Modes, varieties, affordances. In N. Fielding, R.M. Lee, & G. Blank (Eds.), The SAGE Handbook of Online Research Methods (pp. 257-270). Thousand Oaks, CA: SAGE Publications.

Hmelo-Silver, C.E. (2004). Problem-based learning: What and how do students learn? Educational Psychology Review, 16(3), 235-266.

Illich, I. (1971). Deschooling society. New York, NY: Harper & Row.

ITEA. (2007). Standards for technological literacy: Content for the study of technology (3rd ed.). Reston, Virginia: International Technology Education Association.

Ito, M. (2009). Engineering play: A cultural history of children's software. Cambridge, MA: MIT Press.

Ito, M., Baumer, S., Bittanti, M., boyd, d., Cody, R., Herr, B., et al. (2009). Hanging out, messing around, geeking out: Living and learning with new media. Cambridge, MA: MIT Press.

Jenkins, H., Purushotma, R., Weigel, M., Clinton, K., & Robison, A. (2006). Confronting the challenges of participatory culture: Media education for the 21st century. Cambridge, MA: MIT Press.

Kafai, Y.B. (1995). Minds in play: Computer game design as a context for children's learning. Hillsdale, NJ: Lawrence Erlbaum.

Kafai, Y.B. (2006). Constructionism. In K. Sawyer (Ed.), Cambridge Handbook of the Learning Sciences (pp. 35-46). Cambridge: Cambridge University Press.

Kafai, Y.B., & Resnick, M. (Eds.) (1996). Constructionism in practice: Designing, thinking, and learning in a digital world. New Jersey: Lawrence Erlbaum.

Kay, A. (2005). Squeak Etoys, children & learning. VPRI Research Note RN-2005-001. Retrieved August 19, 2012, from http://www.vpri.org/pdf/rn2005001_learning.pdf

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming languages for novice programmers. ACM Computing Surveys, 37(2).

Kirschner, P. & Lai, K. (2007). Online communities of practice in education. Technology, Pedagogy and Education, 16(2), 127-131.

Kirschner, P.A., Sweller, J., & Clark, R.E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. Educational Psychologist, 41(2), 75-86.

Klawe, M., Whitney, T., & Simard, C. (2009). Women in computing – Take 2. Communications of the ACM, 52(2), 68-76.

Kohn, A. (1999). Punished by rewards: The trouble with gold stars, incentive plans, A's, praise, and other bribes. New York, NY: Houghton Mifflin.

Kollock, P., & Smith, M. (1996). Managing the virtual commons: Cooperation and conflict in computer communities. In S.C. Herring (Ed.), Computer-Mediated Communication: Linguistic, Social and Cross-Cultural Perspectives (pp. 109-128). Philadelphia, PA: John Benjamins Publishing Co.

Kolodner, J., Camp, P., Crismond, D., Fasse, B., Gray, J., Holbrook, J., et al. (2003). Problem-based learning meets case-based reasoning in the middle-school science classroom: Putting Learning by Design into practice. Journal of the Learning Sciences, 12(4), 495-547.

Koschmann, T., Kelson, A., Feltovich, P., & Barrows, H. (1996). Computer-supported problem-based learning: A principled approach to the use of computers in collaborative learning. In T. Koschmann (Ed.), CSCL: Theory and practice of an emerging paradigm (pp. 83-124). Hillsdale, NJ: Lawrence Erlbaum.

Krajcik, J.S., & Blumenfeld, P.C. (2006). Project-based learning. In K. Sawyer (Ed.), Cambridge Handbook of the Learning Sciences (pp. 317-334). Cambridge: Cambridge University Press.

Kvale, S. (1996). InterViews: An introduction to qualitative research interviewing. Thousand Oaks, CA: SAGE Publications.

Lankshear, C. & Knobel, M. (2002). Do we have your attention? New literacies, digital technologies, and the education of adolescents. In D.E. Alvermann (Ed.), Adolescents and literacies in a digital world (pp. 19-39). New York, NY: Peter Lang.

Lave, J., & Wenger, E. (1991). Situated learning: Legitimate peripheral participation. New York, NY: Cambridge University Press.

LeCompte, M.D., & Schensul, J.J. (1999). Designing & conducting ethnographic research: Ethnographer's toolkit. Plymouth: AltaMira Press.

Lin, H.T., & Fishman, B. (2009). Scaffolding teacher adaptation by making design intent explicit. In A. Dimitracopoulou, C. O'Malley, D.D. Suthers, & P. Reiman (Eds.), Conference on Computer-Supported Collaborative Learning (Vol. 2, pp. 159-161). Rhodes, Greece: The International Society of the Learning Sciences.

Lorenz, M.O. (1905). Methods of measuring the concentration of wealth. Publications of the American Statistical Association, 9(70), 209-219.

Maeda, J. (2001). Design by numbers. Cambridge, MA: MIT Press.

Maillet, K., & Porta, M. (2010). Consequences of the declining interest in computer science studies in Europe. Education Engineering, April 2010, 71-76.

Manovich, L. (2012). Trending: The promises and the challenges of big social data. In M.K. Gold (Ed.), Debates in the Digital Humanities (pp. 460-475). Minneapolis, MN: The University of Minnesota Press.

Martin, J. (2004). Self-regulated learning, social cognitive theory, and agency. Educational Psychologist, 39(2), 135-145.

Marwick, A.E., & boyd, d. (2011). The drama! Teen conflict, gossip, and bullying in networked publics. Paper presented at A Decade in Internet Time: Symposium on the Dynamics of the Internet and Society. Retrieved September 10, 2012, from http://ssrn.com/abstract=1926349

Mayer, R. (2004). Should there be a three-strikes rule against pure discovery learning? The case for guided methods of instruction. American Psychology, 59(1), 14-19.

Miles, M.B., & Huberman, A.M. (1994). Qualitative data analysis: An expanded sourcebook (2nd ed.). Thousand Oaks, CA: SAGE Publications.

Monroy-Hernandez, A. (2012). Designing for remixing: Supporting an online community of amateur creators. Doctoral dissertation, Massachusetts Institute of Technology.

Monroy-Hernandez, A., Hill, B.M., Gonzalez-Rivero, J., & boyd, d. (2011). Computers can't give credit: How automatic attribution falls short in an online remixing community. Proceedings of the ACM CHI Conference on Human Factors in Computing Systems, 2011.

National Research Council. (1999). Being fluent with information technology. Washington, DC: National Academy Press.

National Research Council. (2010). Report of a workshop on the scope and nature of computational thinking. Washington, DC: National Academy Press.

Newman, M.E.J. (2005). Power laws, Pareto distributions and Zipf's law. Contemporary Physics, 46(5), 323-351.

Paglin, M. (1975). The measurement and trend of inequality: A basic revision. The American Economic Review, 65(4), 598-609.

Palfrey, J., & Gasser, U. (2008). Born digital: Understanding the first generation of digital natives. New York, NY: Basic Books.

Papert, S. (n.d.). Hard fun. Retrieved August 2, 2012, from http://www.papert.org/articles/HardFun.html

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. New York, NY: Basic Books.

Papert, S. (1993). The children's machine: Rethinking school in the age of the computer. New York, NY: Basic Books.

Pausch, R., Burnette, T., Capeheart, A.C., Conway, M., Cosgrove, D., DeLine, R., et al. (1995). Alice: Rapid prototyping system for virtual reality. IEEE Computer Graphics and Applications, 15(3), 8-11.

Pea, R. (2004). The social and technological dimensions of scaffolding and related theoretical concepts for learning, education, and human activity. Journal of the Learning Sciences, 13(3), 423-451.

Pellegrino, J.W., & Hilton, M.L. (Eds.). (2012). Education for life and work: Developing transferable knowledge and skills in the 21st century. Washington, DC: National Academies Press.

Perkins, D. (1986). Knowledge as design. Hillsdale, NJ: Lawrence Erlbaum.

Perkins, D. (1991). Technology meets constructivism: Do they make a marriage? Educational Technology, May, 18-23.

Piaget, J. (1929). The child's conception of the world (J. Tomlinson & A. Tomlinson, Trans.). Maryland: Rowman & Littlefield. (Translation published 2007)

Prensky, M. (2001). Digital natives, digital immigrants. On the Horizon, 9(5), 1-6.

Rainio, A. (2008). From resistance to involvement: Examining agency and control in a playworld activity. Mind, Culture, and Activity, 15(2), 115-140.

Razzouk, R., & Shute, V. (2012). What is design thinking and why is it important? Review of Educational Research, 82(3), 330-349. doi:10.3102/0034654312457429

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: Programming for all. Communications of the ACM, 52(11), 60-67.

Resnick, M., & Silverman, B. (2005). Some reflections on designing construction kits for kids. Proceedings of Interaction Design and Children conference, Boulder, CO.

Rheingold, H. (2012). Net smart: How to thrive online. Cambridge, MA: MIT Press.

Rogers, C.R. (1961). On becoming a person: A therapist's view of psychotherapy. New York, NY: Houghton Mifflin.

Rogoff, B. (1994). Developing understanding of the idea of communities of learners. Mind, Culture and Activity, 1(4), 209–229.

Roschelle, J., Bakia, M., Toyama, Y., & Patton, C. (2011). Eight issues for learning scientists about education and the economy. Journal of the Learning Sciences, 20(1), 3-49.

Rosenbaum, E. (2009). Jots: Cultivating reflective learning in Scratch. Master's thesis, Massachusetts Institute of Technology.

Roque, R. (2012). Making together: Creative collaboration for everyone. Master's thesis, Massachusetts Institute of Technology.

Rushkoff, D. (2010). Program or be programmed: Ten commands for a digital age. New York, NY: OR Books.

Sawyer, R.K. (2004). Improvised lessons: Collaborative discussion in the constructivist classroom. Teaching Education, 15(2), 189-201.

Sawyer, K. (2006). Explaining creativity: The science of human innovation. Oxford: Oxford University Press.

Scardamalia, M., & Bereiter, C. (1991). Higher levels of agency for children in knowledge building: A challenge for the design of new knowledge media. Journal of the Learning Sciences, 1(1), 37-68.

Schlager, M., & Fusco, J. (2003). Teacher professional development, technology, and communities of practice: Are we putting the cart before the horse? Information Society, 19(3), 203–220.

Schofield, J.W. (1995). Computers and classroom culture. Cambridge: Cambridge University Press.

Schön, D. (1983). The reflective practitioner: How professionals think in action. New York, NY: Basic Books.

Schwartz, D.L., & Okita, S. (2009). The productive agency in learning by teaching. Unpublished manuscript. Retrieved December 30, 2011, from http://aaalab.stanford.edu/papers/Productive_Agency_in_Learning_by_Teaching.pdf

Seiter, E. (2008). Practicing at home: Computers, pianos, and cultural capital. In T. McPherson (Ed.), Digital youth, innovation, and the unexpected (pp. 27-52). Cambridge, MA: MIT Press. doi:10.1162/dmal.9780262633598.027

Selwyn, N. (2009). The digital native – myth and reality. New Information Perspectives, 61(4), 364-379.

Sewell, W.H. (1992). A theory of structure: Duality, agency, and transformation. American Journal of Sociology, 98(1), 1-29.

Shirky, C. (2003). Power laws, weblogs, and inequality. Retrieved September 5, 2012, from http://www.shirky.com/writings/powerlaw_weblog.html

Skinner, B.F. (1986). Programmed instruction revisited. The Phi Delta Kappan, 68(2), 103-110.

Stake, R.E. (2010). Qualitative research: Studying how things work. New York, NY: Guilford Press.

Tapscott, D. (2008). Grown up digital: How the net generation is changing your world. New York, NY: McGraw-Hill.

The Design-Based Research Collective. (2003). Design-based research: An emerging paradigm for educational inquiry. Educational Researcher, 32(1), 5-8.

The Partnership for 21st Century Skills. (2009). Learning for the 21st century. Retrieved March 1, 2012, from http://www.p21.org/storage/images/stories/otherdocs/p21up_Report.pdf

Thomas, D., & Brown, J.S. (2011). A new culture of learning: Cultivating the imagination for a world of constant change. CreateSpace.

Tough, P. (2012). How children succeed: Grit, curiosity, and the hidden power of character. New York, NY: Houghton Mifflin Harcourt.

Turkle, S., & Papert, S. (1990). Epistemological pluralism: Styles and voices within the computer culture. Signs, 16(1), 128-157.

Ullman, E. (August 8, 2012) Errant code? It's not just a bug. Retrieved August 8, 2012, from http://www.nytimes.com/2012/08/09/opinion/after-knight-capital-new-code-for-trades.html

U.S. Department of Education, Office of Educational Technology. (2010). Transforming American education: Learning powered by technology. Retrieved June 10, 2012, from http://www.ed.gov/sites/default/files/netp2010-execsumm.pdf

Van Maanen, J. (1988). Tales of the field: On writing ethnography. Chicago: The University of Chicago Press.

Vaughan, N. (2004). Technology in support of faculty learning communities. New Directions for Teaching and Learning, 97, 101–109.

Vygotsky, L.S. (1978). Mind in society: Development of higher psychological processes (14th ed.). Cambridge, MA: Harvard University Press.

Wenger, E. (1998). Communities of practice: Learning, meaning, and identity. Cambridge: Cambridge University Press.

Wenger, E., McDermott, R., and Snyder, W. M. (2002). Cultivating communities of practice. Cambridge, MA: Harvard Business School Press.

WGBH, & ACM. (2009). New image for computing: Report on market research. Retrieved March 1, 2012, from http://www.dotdiva.org/downloads/New_Image_Computing_final.pdf

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

Winnicott, D.W. (1971). Playing and reality. London: Tavistock Publications.

Wiske, M.S., Perkins, D., & Spicer, D.E. (2006). Piaget goes digital: Negotiating accommodation of practice to principles. In C. Dede (Ed.), Online professional development for teachers (pp. 49-67). Cambridge, MA: Harvard Education Press.

Yin, R.K. (2009). Case study research: Design and methods (4th ed.). Thousand Oaks, CA: SAGE Publications.

Appendix A: Quantitative Participation Data

The following tables provide detailed quantitative information about the participation of the 30 kids who were interviewed, as well as quantitative information about "active" Scratch community participants, for comparison. Active participants are the 75,568 Scratch community members (6.2% of 1,222,242) who have posted four or more Scratch projects and have been active in the online community for at least 28 days.

Table A.1, in addition to listing how long accounts had been active at the time of the interview, focuses on technical aspects of participation: the number of projects created, the total number of scripts written (across all projects, average number of scripts per project, maximum number of scripts in any project), and the number of projects that were remixes of others' work. Table A.2 focuses on social aspects of participation: the number of people that the Scratcher identified as a friend, the number of people who identified the Scratcher as a friend, the number of "love-its" that the Scratcher had given to others, and the number of love-its that the Scratcher had received (total across all projects, average love-its per project, maximum love-its on any project). Table A.3 focuses on social exchanges through text: the number of comments given, the number of comments received (across all projects, average number of comments per project, maximum number of comments on any project), and the number of posts contributed to the Scratch online forums.

Table A.1 Summary of account statistics (technical) for the out-of-school interviewees.

Group	Name	Account (years)	Projects	Total Scripts	Avg Scripts Per Project	Max Scripts Per Project	Remixes
A	Allison	0.15	2	5	2.5	3	0
A	Adele	0.09	5	44	8.8	35	0
A	Robin	0.07	8	148	18.5	59	4
A	Evan	0.10	6	79	13.2	25	0
B	Jackson	0.02	20	435	21.8	58	8
B	Jenson	1.96	21	598	28.5	176	1
B	Brook	0.65	34	270	7.9	91	5
B	Connor	1.00	36	397	11.0	50	4
B	Easton	1.88	70	1311	18.7	121	14
B	Edgar	2.76	150	2648	17.7	265	20
B	Brent	2.43	175	977	5.6	46	5
C	Sebastian	0.41	10	213	21.3	74	0
C	Aaron	1.09	257	4472	17.4	179	44
C	Devon	0.59	65	957	14.7	218	20
C	Barry	2.97	227	4387	19.3	286	41
C	Matt	1.04	90	2530	28.1	154	19
C	Lindsey	1.72	150	5477	36.5	669	42
C	Ashleigh	0.92	64	2248	35.1	607	13
C	Nevin	0.29	236	1554	6.6	120	105
C	Fletcher	1.82	147	3138	21.3	265	10
C	David	1.88	307	2666	8.7	147	88
D	Chuck	1.05	31	1348	43.5	242	1
D	Sonia	1.73	54	941	17.4	74	6
D	Chelsey	0.94	128	1577	12.3	72	6
D	Lori	0.27	169	3127	18.5	536	0
D	Bradley	1.60	339	3788	11.2	988	77
D	Clark	3.54	221	9937	45.0	918	55
D	Jan	4.44	43	947	22.0	108	9
D	Eva	2.15	324	5990	18.5	221	61
D	Lana	4.77	192	5095	26.5	343	19
A	50 Percentile	0.10	6	62	11.0	30	0
B	50 Percentile	1.88	36	598	17.7	91	5
C	50 Percentile	1.07	149	2598	20.3	199	31
D	50 Percentile	1.73	169	3127	18.5	242	9
Pop	0 Percentile	0.00	4	0	0.0	0	0
Pop	25 Percentile	0.24	6	47	4.9	14	0
Pop	50 Percentile	0.67	11	133	10.4	38	1
Pop	75 Percentile	1.59	24	417	22.1	101	4
Pop	90 Percentile	2.70	59	1133	43.6	208	12
Pop	95 Percentile	3.36	104	1994	64.6	308	24
Pop	97 Percentile	3.81	151	2841	84.3	402	37
Pop	99 Percentile	4.52	286	5474	141.5	704	79
Pop	100 Percentile	5.55	6374	438998	8457.5	436214	746
Pop	Average	1.08	28	517	20.0	120	6

Table A.2 Summary of account statistics (social) for the out-of-school interviewees.

Group	Name	Friends	Friended By	Loveits Given	Total Loveits Received	Avg Loveits Received	Max Loveits Received
A	Allison	0	0	1	1	0.5	1
A	Adele	0	1	1	0	0.0	0
A	Robin	1	3	12	3	0.4	1
A	Evan	2	1	12	3	0.5	2
B	Jackson	3	4	7	3	0.2	2
B	Jenson	2	3	9	10	0.5	2
B	Brook	2	2	6	3	0.1	1
B	Connor	7	12	53	30	0.8	4
B	Easton	16	14	195	28	0.4	9
B	Edgar	0	9	27	31	0.2	4
B	Brent	4	13	51	30	0.2	5
C	Sebastian	5	11	92	66	6.6	21
C	Aaron	26	55	186	129	0.5	10
C	Devon	354	183	2586	157	2.4	16
C	Barry	198	116	229	84	0.4	12
C	Matt	13	79	106	188	2.1	134
C	Lindsey	5	99	152	174	1.2	54
C	Ashleigh	25	153	748	594	9.3	157
C	Nevin	85	151	604	244	1.0	29
C	Fletcher	78	144	77	271	1.8	26
C	David	675	244	1831	104	0.3	16
D	Chuck	72	340	54	398	12.8	122
D	Sonia	26	154	257	537	9.9	49
D	Chelsey	11	236	176	277	2.2	58
D	Lori	4	170	54	519	3.1	97
D	Bradley	86	488	326	1189	3.5	303
D	Clark	194	562	7	1074	4.9	114
D	Jan	125	675	321	2016	46.9	646
D	Eva	12	306	330	1414	4.4	699
D	Lana	700	1576	1457	2387	12.4	232
A	50 Percentile	1	1	7	2	0.4	1
B	50 Percentile	3	9	27	28	0.2	4
C	50 Percentile	52	130	208	166	1.5	24
D	50 Percentile	72	340	257	1074	4.9	122
Pop	0 Percentile	0	0	0	0	0.0	0
Pop	25 Percentile	0	0	0	0	0.0	0
Pop	50 Percentile	1	2	1	1	0.1	1
Pop	75 Percentile	6	8	9	7	0.4	2
Pop	90 Percentile	23	29	40	32	1.0	6
Pop	95 Percentile	54	61	93	80	1.8	13
Pop	97 Percentile	89	98	152	142	2.6	23
Pop	99 Percentile	223	214	381	400	6.2	83
Pop	100 Percentile	4283	3200	7091	18676	200.2	4086
Pop	Average	13	14	22	24	0.5	5

Table A.3 Summary of account statistics (social, through text) for the out-of-school interviewees.

Group	Name	Comments Given	Total Comments Received	Avg Comments Received	Max Comments Received	Forum Posts
A	Allison	0	0	0.0	0	0
A	Adele	0	1	0.2	1	0
A	Robin	29	5	0.6	3	0
A	Evan	11	7	1.2	5	0
B	Jackson	11	12	0.6	2	0
B	Jenson	13	18	0.9	4	0
B	Brook	39	21	0.6	6	3
B	Connor	97	32	0.9	7	0
B	Easton	188	62	0.9	8	3
B	Edgar	121	115	0.8	9	2
B	Brent	77	155	0.9	87	2
C	Sebastian	149	150	15.0	62	86
C	Aaron	875	436	1.7	26	1
C	Devon	1998	463	7.1	51	733
C	Barry	1292	550	2.4	72	1562
C	Matt	226	618	6.9	419	2
C	Lindsey	1093	908	6.1	151	156
C	Ashleigh	1803	1203	18.8	232	20
C	Nevin	1846	1210	5.1	165	187
C	Fletcher	688	1302	8.9	180	1367
C	David	3369	1605	5.2	55	25
D	Chuck	157	1221	39.4	460	18
D	Sonia	642	1648	30.5	183	1259
D	Chelsey	1098	1660	13.0	336	172
D	Lori	419	2286	13.5	145	329
D	Bradley	3077	3198	9.4	423	2075
D	Clark	2821	3901	17.7	211	5052
D	Jan	797	4163	96.8	882	2045
D	Eva	6272	5801	17.9	599	38
D	Lana	5655	10481	54.6	466	5295
A	50 Percentile	6	3	0.4	2	0
B	50 Percentile	77	32	0.9	7	2
C	50 Percentile	1193	763	6.5	112	121
D	50 Percentile	1098	3198	17.9	423	1259
Pop	0 Percentile	0	0	0	0	0
Pop	25 Percentile	0	1	0.1	1	0
Pop	50 Percentile	4	6	0.5	2	0
Pop	75 Percentile	31	27	1.5	7	0
Pop	90 Percentile	164	133	3.7	21	2
Pop	95 Percentile	416	340	6.5	49	9
Pop	97 Percentile	729	622	9.6	89	23
Pop	99 Percentile	1722	1753	19.9	237	202
Pop	100 Percentile	31292	55840	416	5719	16468
Pop	Average	95	100	1.8	14	15

Appendix B: Interview Themes

The following tables illustrate the quantitative density of the qualitatively thematized interview data. (The interview data was later supplemented with observation and artifact data based on this thematizing.) The themes are sorted by order of appearance in Chapter 4 (Table B.1) and Chapter 5 (Table B.2), respectively.

As described in Chapter 3, the themes were initially developed using an emic approach, drawing on language used by kids and teachers to describe their experiences. Then, through an iterative coding/clustering/grouping process, the themes were refined and organized using a narrative structure of goals, challenges to those goals, strategies for overcoming challenges, and processes of validation as an organizing strategy. Table B.1 and Table B.2 illustrate the mapping from the high-level themes that organize Chapter 4 and Chapter 5 to the particular codes from the interviews. The *Respondents* column reports the number of respondents whose transcripts were coded with this theme (maximum 30), and the *Instances* column reports the number of coded instances across all interviews. Themes without respondent and instance values were used for organizational purposes, and not uniquely coded.

Table B.1 Quantitative density of qualitative themes presented in Chapter 4 (Kids).

Theme	Respondents	Instances
Enjoying freedom		
what Scratch is - freedom	23	33
what Scratch is good for - freedom	22	44
learner freedom	13	32
freedom of creation		
freedom in creating	19	25
where they get ideas		
getting ideas - other sources of inspiration	17	22
getting ideas - from others	15	17
getting ideas - from projects	14	17
getting ideas - from personal interests	12	27
beyond Scratch - hobbies and interests	21	23
freedom of process		
different from school experiences		
school - Scratch experiences	24	54
school - technology experiences	16	22
school - learning experiences	6	8
different from home experiences		
home - Scratch experiences	14	16
different from other tech experiences		
beyond Scratch - other technology use	15	22
beyond Scratch - programming	7	8
compared to other languages	13	20
Getting stuck		
programming is challenging	11	15
getting stuck	14	20
Making progress		
getting better	26	50
improving work	10	11
individual strategies		
learning alone	14	20
experimenting		
learning by experimenting	4	4
experimenting	14	19
debugging	15	24
testing	6	13

Table B.1 continued.

Theme	Respondents	Instances
iterative, incremental	11	19
planning		
planning	16	25
organizing	5	5
breaking down the problem	8	14
optimizing	2	8
multiple accounts	6	7
compromising		
compromising, reframing, approximation	10	13
persevering		
time, practice	22	33
persistence	10	17
taking a break		
giving up	15	22
taking a break	7	9
social strategies		
what Scratch is - community	11	14
learning with others	7	8
asking for help		
access to expertise at home	14	16
learning from others	18	31
getting help from other	19	40
studying projects		
learning from examples	15	29
looking at examples	7	9
style	1	1
remixing work		
framings of remixing		
customizing	15	18
improving	6	6
way to learn	6	6
building blocks	5	6
remixing		
learning opportunities	6	6
challenge, copying	17	20
challenge, credit	5	6

Table B.1 continued.

Theme	Respondents	Instances
working with others		
collaboration	25	38
collaboration, challenges	9	14
helping others		
helping others	28	52
Finding audience		
motivation		
receiving encouragement to create	12	12
getting feedback from others to improve work	10	14
wanting to engage others	7	9
displaying work to a broad audience	7	7
interests, shared	7	12
friends	7	8
challenges		
comments - receiving	17	27
comments - giving	16	20
comments - reciprocity	4	5
negative interactions		
interests, different or conflicting	6	8
meeting people in person	4	7
not connecting with people	3	4
mean or unproductive comments	9	13
inappropriate projects	4	4
lying	3	3
hacking accounts	2	2
popularity		
importance of	14	15
challenge of attaining	17	25
strategies for attaining	12	20
downsides	4	4

Table B.2 Quantitative density of qualitative themes presented in Chapter 5 (Teachers).

Theme	Respondents	Instances
Supporting problem-solvers		
motivation - fostering creativity	16	17
motivation - understanding computers, technology	12	16
motivation - developing problem-solving abilities	7	10
motivation - expressing ideas in an another medium	7	9
motivation - meaningful computer activity	6	7
motivation - experiencing programming	5	7
motivation - developing logic	5	6
motivation - experiencing computer science	2	2
motivation - broadening participation	1	1
motivation - contextualizing abstract knowledge	1	1
motivation - learning about learning	1	1
Negotiating open-endedness		
knowing how much structure to provide		
building capacity	11	18
feeling tension between freedom and structure	9	14
students need time	9	12
maintaining control	8	10
deciding what examples to show	1	2
lack of time in a teacher's schedule	13	19
working with large numbers of students	6	8
fears		
not knowing (enough about Scratch)	14	33
lacking understanding of what Scratch is	9	13
feeling isolated	7	7
not being able to keep up with tech	5	5
challenges, negotiating different levels of experience		
pedagogical challenge	5	7
keeping more advanced Scratchers engaged	6	7
discourages newer Scratchers	4	4
unevenness in group work	2	2
challenges, negotiating student attitudes and anxieties		
resistance to collaboration or sharing or remixing	9	9
frustration with not being able to do what they want	6	6
resistance to experimenting	4	5
unmotivated learners	3	5

Table B.2 continued.

Theme	Respondents	Instances
keeping kids focused on work	3	3
disinterest in computers and technology	2	2
helping kids who get stuck		
inappropriate scoping of project	6	8
lack of persistence	4	7
uncertainty about what the problem is	3	3
on bug in project	2	2
not knowing what to work on	1	3
Building culture		
create expectations of learning culture	3	6
advice from teachers		
try it yourself		
try it out yourself	17	29
connect with other educators	14	26
accumulate pointers to great resources	13	16
follow their interests		
connect to learner interests	16	29
understand that there are multiple approaches	12	13
create multiple pathways of participation	3	4
develop interesting project ideas and activities	11	13
be a guide		
provide metacognitive support		
help students break it down	15	19
be a consultant or coach	12	12
help kids develop design strategies	7	13
ask questions of students	7	10
support reflection	7	7
provide more explicit guidance when needed	11	16
encourage		
encourage kids to explore	12	22
encourage kids to solve problems	11	20
encourage kids' autonomy	9	13
encourage kids to challenge themselves	9	11
encourage kids' experimenting	9	10
encourage kids' creativity	7	7
encourage kids' fearlessness	3	3

Table B.2 continued.

Theme	Respondents	Instances
teacher attitudes		
be enthusiastic	7	9
be sympathetic to kids' experiences	3	4
be a model learner	4	7
be curious	3	3
feel OK with not knowing everything		
feel OK about not knowing everything	11	13
learn from student expertise	6	6
learn w/students by working together	7	7
say you do not know	3	3
avoid showing everything	2	2
create opportunities to share		
support peer learning	18	30
have students work together on projects	11	18
recruit student mentors or helpers	7	13
have students present their projects	7	9
invite an outside audience		
other teachers, to inspire	6	8
parents and other family	6	7
broader community	5	5
use the website	9	11
have students share with a neighbor	2	3
have students do demos	2	2
Legitimizing learning		
setting of Scratch use	22	29
challenges, school culture - going against the way school is done	16	20
challenges, school culture - performance or perception pressure	12	17
challenges, school culture - curricular connections	4	5
knowing how to assess		
creative work	10	14
acknowledging effort, engagement, enjoyment	6	8
technical understanding	5	6
using the web		
inappropriate content or behavior	5	6
worrying about copying	3	4
keeping things related to Scratch	1	1

Table B.2 continued.

Theme	Respondents	Instances
wanting other venues for sharing Scratch projects	1	1
lacking instructional support and resources		
lack of access, to computers or other tech resources		
at school	10	13
at home	2	2
finding funding	6	8
lacking quality instructional resources	2	2