Online Learning Webs:
Designing Support Structures for Online Communities

by
Champika Fernando

B.S. Computer Science, University of Toronto (2006)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
September 2014
© Massachusetts Institute of Technology 2014. All rights reserved.
Abstract
This thesis explores how we can design online learning communities to better support connections to the people and resources beginners need when learning to program. I describe and analyze the design and implementation of the Scripts Workshop, a learning environment that supports members of the Scratch online community who are stuck on a programming problem in a Scratch project. The Scripts Workshop considers the People, Activities and Spaces needed to support these users in getting un-stuck. I conclude by describing a set of design principles for building learning webs within online communities, derived from the Scripts Workshop experiment.

Thesis Supervisor: Mitchel Resnick
Title: LEGO Papert Professor of Learning Research, Program in Media Arts and Sciences
Online Learning Webs:
Designing Support Structures for Online Communities

by
Champika Fernando

The following people served as readers for this thesis:

Thesis Reader

Pattie Maes
Alex W. Dreyfoos Professor of Media Technology
Program in Media Arts and Sciences, MIT

Karen Brennan
Assistant Professor
Graduate School of Education, Harvard University

Justin Reich
Richard L. Menschel HarvardX Research Fellow
Office of the President and Provost, Harvard University
Considering the focus of this thesis is on the people and resources that support learning, it seems particularly pertinent to begin by thanking all the people who supported me as I struggled through this very foreign process of writing a thesis.

Thank you first to my patient and ever-supportive advisor Mitch Resnick. So much of what I’ve learned over the past few years I’ve learned through conversations and collaborative design meetings with Mitch. Mitch – thank you so much for being a source of inspiration, support, and friendship.

Thank you to my readers Pattie Maes, Karen Brennan, and Justin Reich, who not only tolerated my tardiness, but also took the time to give me thoughtful and constructive feedback in spite of it. Thank you all for being so generous with your time and advice throughout the process and particularly in the final stretch.

Thank you to the Scripts Workshop design team Ricarose Roque, Sarah Otts, Mark Goff, Eric Schilling, and Amos Blanton whose work this thesis is based on. Weekly meetings have never been so fun.

Ricarose Roque, who in addition to leading the Scripts Workshop team, provided invaluable advice, resources, moral support, and friendship along the way. Thank you Ricarose!

Thank you to the Scratch Mentors and Script Helpers who participated in the Scripts Workshop. And, to all the members of the Scratch community – you never cease to surprise and inspire me.

Stephanie Gayle – thank you for not letting me give up and for giving me that stack of thesis quality paper months ago – it came in handy.

To the members of the Scratch Team and the Lifelong Kindergarten research group – your kindness, creativity, and passion have inspired me and I feel so fortunate to be a part of this community.
# Contents

1 Introduction ................................................................................................................................. 9  
   1.1 Thesis Overview ......................................................................................................................... 12  

2 Theoretical Inspirations ............................................................................................................. 14  
   2.1 Learning Webs .............................................................................................................................. 14  
   2.2 Constructionism .......................................................................................................................... 16  
   2.3 Social Perspectives on Learning .................................................................................................. 18  

3 Design Inspirations ...................................................................................................................... 19  
   3.1 Scratch Online Community .......................................................................................................... 19  
   3.2 Stack Overflow ............................................................................................................................ 23  

4 Existing Scratch Support Spaces ............................................................................................... 26  
   4.1 Scratch Team Created Resources ............................................................................................... 27  
   4.2 Scratch Team Designed, Community Supported Spaces ......................................................... 28  
   4.3 Community Designed, Community Supported Spaces ............................................................. 30  
   4.4 Limitations .................................................................................................................................. 32  

5 Scripts Workshop ....................................................................................................................... 34  
   5.1 People ....................................................................................................................................... 34  
   5.2 Activities ................................................................................................................................... 41  
   5.3 Spaces ...................................................................................................................................... 49  
   5.4 Possibilities and Limitations ...................................................................................................... 59  

6 Implications: Scratch Community ............................................................................................... 62  
   6.1 Identity ..................................................................................................................................... 62  
   6.2 Communication ........................................................................................................................... 64  
   6.3 Diversity of Interest .................................................................................................................... 66  

7 Implications: Beyond Scratch ...................................................................................................... 68
7.1 Relationships and Inspiration ........................................................................................................... 68
7.2 Technologies for communication ........................................................................................................ 68
7.3 Policy .............................................................................................................................................. 69
7.4 Massive Open Online Courses ......................................................................................................... 69
7.5 Internet as Learning Web .................................................................................................................. 70

Bibliography ......................................................................................................................................... 71
List of Figures

Figure 1-1: (left) Google search results for “Learn to code”, (right) screenshots from online programming apps (Codeacademy, Code Avengers, Code Combat) .................................................. 10
Figure 1-2: (left) Google search results for “Programming help”. (right) Homepage of the popular Stack Overflow programming Q&A community .................................................. 11
Figure 3-1: The Scratch online community (behind) and the Scratch programming editor (front) .......................................................................................................................... 20
Figure 3-2: An example of comments on a Scratch Project .................................................. 21
Figure 3-3: The backpack within the Scratch editor for moving assets between projects .... 21
Figure 3-4: Scratch discussion forums (back) and an example of a Scratch studio (front) .. 22
Figure 3-5: Scratch community guidelines ........................................................................ 23
Figure 3-6: The Stack Overflow homepage ....................................................................... 24
Figure 4-1: An illustration of the Scratch resource ecosystem ........................................... 27
Figure 4-2: Scratch help section (left) and in-editor tip-bar (right) .................................... 28
Figure 4-3: Scratch Welcoming Committee ...................................................................... 29
Figure 4-4: Examples of the variety of types of Scratcher created tutorials ...................... 31
Figure 4-5: Scratcher created community of tutorials and helpers .................................. 31
Figure 4-6: Comments from Scratchers wanting to contribute to the Tutorial Madness studio .......................................................................................................................... 32
Figure 5-1: Comments within the Scripts Workshop reflect people looking to get help .... 35
Figure 5-2: Comments within a typical “Helper Studio” reflect people looking to provide help .................................................................................................................................. 36
Figure 5-4: Script Helpers discussing importance of quick help ...................................... 37
Figure 5-5: Unsupportive interaction between a Script Helper and Scratcher .................... 39
Figure 5-6: Scratch Mentor modeling supportive help ....................................................... 40
Figure 5-7: Scratch Mentor providing feedback on how to improve a tutorial project ...... 43
Figure 5-8: Creative tutorial project, provides two sprites, and suggests you write the script for the third ....................................................................................................................... 44
Figure 5-9: Positive remarks from community on a Scripts Workshop tutorial project ...... 45
Figure 5-10: Scratcher looking for help is left hanging in the discussion forums ............. 46
Figure 5-11: In providing support a Script Helper points to another Scratcher who may be useful .......................................................................................................................... 47
Figure 5-12: Script Helpers responding poorly to a request for help .................................. 48
Figure 5-13: Scratch Mentor modeling supportive help ..................................................... 49
Figure 5-14: Main Scripts Workshop space with listing of projects looking for help .......... 50
Figure 5-15: Scratch Team and Scratch Mentors participating in the Scripts Workshop, as listed in the ‘curators’ tab of the Scripts Workshop ........................................................................ 50
Figure 5-16: Script Helpers as listed in the ‘curators’ tab of the Scripts Workshop .......... 51
Figure 5-17: A Scratcher looking for help in the Scripts Workshop .................................. 52
Figure 5-18: The same Scratcher from figure 5:17, now providing help in the Scripts Workshop .......................................................................................................................... 52
Figure 5-19: Example of a Script Helper proving help on a Scratchers project.......................... 54
Figure 5-20: Advice from a Scratch Mentor on how to provide help............................................ 55
Figure 5-21: Scratch Mentor describing challenges of online support............................................. 55
Figure 5-22: Scratch Mentor describing challenges of programming support................................. 56
Figure 5-23: Scratch Mentor describing challenges of pedagogical support.................................. 56
Figure 5-24: ‘What it takes to be a Scratch Helper’: A project that explains the requirements of a Scripts Helper. .......................................................................................................................... 57
Figure 5-25: A chart of individual Script Helpers areas of expertise. Names have been greyed out.............................................................................................................................................. 59
Figure 6-1: Scratch user profile page .................................................................................................. 63
Figure 6-2: Scratch user profile with helper identity information..................................................... 64
Figure 6-3: Backpack for moving sprites, scripts, sounds, and costumes between projects.65
Figure 6-4: A project comment with and without the ability to add scripts..................................... 66
Introduction

Over the past year there has been a groundswell of national interest in teaching and learning computer programming. This is illustrated by the attention and success garnered by last December’s Hour of Code campaign. The campaign was launched by Code.org (a group backed by the tech industry), and invited students across the United States to spend at least one hour learning to code during Computer Science Education Week 2013. The campaign was promoted by everyone from tech giants like Google and Facebook, to actors and pop stars like Ashton Kutcher and Shakira. Even the President of the United States encouraged students to participate, suggesting, "Don't just buy the latest video game; make it. Don't just download the latest app; help design it" [1]. At the end of the week Code.org reported that 1 in every 5 students in the United States had participated [2].

Participating in the Hour of Code meant completing one or more structured tutorials, which Code.org provided on their site. Most of these were games or puzzles that teach basic concepts and are meant to be done on your own. But the excitement around programming goes well beyond Hour of Code. Today, typing "Learn to code" into Google returns millions of hits. Most of these are geared towards beginners and also focus on tutorials, programming challenges, games, and step-by-step projects (Figure 1-1).
Figure 1-1: (left) Google search results for “Learn to code”, (right) screenshots from online programming apps (Codeacademy, Code Avengers, Code Combat)

For many beginners the structure provided by these activities may be beneficial. But there is a huge leap between these activities and using code, as the President suggests, to create something you've designed and imagined yourself - wherein lies the real power of programming. The problems you run into when working on real-world projects are not as well defined as these puzzles and almost always require reaching out to other people and resources for support. A second search for "Programming help" on Google returns thousands of results that focus on connecting you with communities of 'expert programmers' who can ostensibly quickly answer your programming questions as you work on projects (Figure 1-2).
For professional programmers, community Q&A sites like Stack Overflow are incredible sources of support. However, visiting any one of these sites you quickly realize they aren't spaces designed for beginners or kids. They value efficiency and accuracy, often at the cost of pedagogy, which results in a culture that can be intimidating and unwelcoming to beginners. This feeling is encapsulated in a post from a beginner in the popular programming help community, Stack Overflow:

I'm a relatively new programmer so some things that are simple to many are complex or at least not immediately obvious to me. Whenever I post a question except for one or two responders, most people seem to love giving sarcastic and mocking answers. Why is the community like that? I think there should be perhaps a beginner tag so that the people who don't want to deal with that don't have to. [3]

These two search results illustrate a gap in current online spaces for learning to program. Beginners have an abundance of tools available to them but don't have many clear pathways for connecting to resources and social support as they tackle messy problems on real projects.

**Figure 1-2:** (left) Google search results for “Programming help”. (right) Homepage of the popular Stack Overflow programming Q&A community
This is not a problem unique to online learning spaces. As Brennan [4] points out, support for social dimensions of learning have been underutilized in computing education. People and relationships are inseparable from learning and there has been much research in the learning sciences that points to this [5]. In the design of online learning environments we have an opportunity to rethink how we teach and learn computational thinking to better take advantage of the Internet’s capacity to connect learners with a variety of people and resources for support.

In a 1971 essay titled "Deschooling Society", Ivan Illich suggests that our general approach to designing educational systems is fundamentally flawed.

The planning of new educational institutions must not start with the question, "What should someone learn?" but with the question, "What kinds of things and people might learners want to be in contact with in order to learn?"

In this thesis I explore this question through the design of an online learning environment that facilitates kids connecting with the things and people needed to help them as they work on personally meaningful programming projects. I focus specifically on the context of kids because most of the work being done around social online learning today focuses on adults. I’ll introduce the design and implementation of the Scripts Workshop, a learning environment within the Scratch online community that aims to support kids who are stuck on programming problems by connecting them with resources, skilled peers, and mentors. The overarching goal of this work is to influence the conversation around the design of online learning communities and inspire other designers of these spaces to consider more social approaches to learning.

1.1 Thesis Overview

In Chapter 2, I give an overview of the pedagogical inspiration and theoretical framework for my
thesis. I’ll describe Ivan Illich’s learning webs (Resources, Skill Models, and Mentors), which provide the inspiration for this thesis. I’ll also describe ideas from Constructionism and Social theories of learning that guide this work.

In Chapter 3, I describe the design of two online communities, the Scratch online community and Stack Overflow, which are sources of design inspiration and insight for this thesis. Though I’m critical of Stack Overflow as a community for beginners and kids, there is a lot that can be learned from their design choices in supporting professionals and experts.

In Chapter 4, I survey and describe existing ways that members of the Scratch online community find support. I’ll also explain the limitations of the current options.

In Chapter 5, I describe and analyze the design and implementation of the Scripts Workshop in terms of the People, Activities, and Spaces involved. I’ll conclude with 4 design principles that can be taken from the Scripts Workshop experiment. I’ll also describe some of the limitations of the workshop, which provide the basis for Chapter 6.

In Chapter 6, based on the work in the Scripts Workshop, I propose design changes within the Scratch online community to enable better systems of community support.

In Chapter 7, based on the work in the Scripts Workshop, I propose future work outside the scope of Scratch but within other areas of online learning.
Theoretical Inspirations

“The planning of new educational institutions must not start with the question, ‘What should someone learn?’ but with the question ‘what kinds of things and people might learners want to be in contact with in order to learn?’”
Illich [6]

The above quote describes the overarching pedagogical and design philosophy that guides my thesis. In addition to this networked, social approach to learning, I take a community-based constructionist approach; where people are engaged in making tangible objects within a community of practice. In this chapter I give an overview of the theoretical and pedagogical influences of this work.

1.2 Learning Webs

In support of his goal of connecting learners with the people and things that can help them learn, Illich suggests four networks that learners should have access to.

“The child grows up in a world of things, surrounded by people who serve as models for skills and values. He finds peers who challenge him to argue, to compete, to
cooperate, and to understand; and if the child is lucky, he is exposed to confrontation or criticism by an experienced elder who really cares. Things, models, peers, and elders are four resources each of which requires a different type of arrangement to ensure that everybody has ample access to it. “

Illich [6]

In this thesis, I focus on three of these four networks, which I define below along with the terms I’ll use to refer to them throughout this thesis.

**Resources (Things)**

These are the tools, objects, and spaces that a learner should be in contact with in the process of learning. In terms of spaces, Illich [6] suggests that the “quality of the environment and the relationship of a person to it will determine how much he learns”. In terms of tools and objects, he describes the importance of having artifacts whose inner-workings are apparent and accessible as a source of learning.

**Skilled Peers (Models)**

These are people with more expertise (but not necessarily pedagogical experience) who are available and willing to demonstrate how to do what the learner wants to do. Illich [6] points out that though skill demonstration can be recorded and automated, the real-human aspect is preferable and more effective for many learners. I’ll note that I’m collapsing two networks that Illich makes distinct, “models” and “peers”. He describes the first as “skill teachers” or people with an expertise over the learner. He describes the second as peers who “pursue a common goal based on common interest and ability”. Instead, I focus on peers as skill teachers. In the Scratch online community there exists a range of experience, expertise, within peer groups around particular interests. Therefore, there is potential to see peers not just as peers but also as skill models.
Mentors (Elders)

Illich [6] describes these as people as “professional educators”. Their role is to provide pedagogical guidance in the learning experiences of others. Mentors require a knowledge of human learning, an awareness of educational resources, and ‘wisdom’ based on experience. Though Illich likely imagines this role as one filled by adults, in this thesis, I explore the potential for kids to take on this role based on their experience in the Scratch online community and demonstration of the qualities listed above.

1.3 Constructionism

The Scratch programming language and online community, in which the Scripts Workshop is implemented, are heavily inspired by Seymour Papert’s theory of Constructionism. Papert defines Constructionism in a National Science Foundation proposal entitled *Constructionism: A New Opportunity for Elementary Science Education*.

“From constructivist theories of psychology we take a view of learning as a reconstruction rather than as a transmission of knowledge. Then we extend the idea of manipulative materials to the idea that learning is most effective when part of an activity the learner experiences as constructing a meaningful product.”

Papert [7]

Papert and his colleagues and students at MIT were some of the first to start thinking about how kids could program computers to control robots, compose music, create games and do other creative activities [8]. They designed the LOGO programming language in 1967 as a means for that vision. LOGO allowed children to program a small turtle on the screen. Since then, it has served as a model for many other educational programming languages including Scratch.

In this thesis I take inspiration from the following “big ideas behind constructionism” as outlined
by Papert [9]:

**Learning by Doing**

Papert suggests we learn better when learning is part of doing something we find interesting. Taking inspiration from this idea, in Scratch, instead of working on programming problems or puzzles, kids learn to program as they create personally meaningful projects. When working on programming projects like these, one invariably runs into difficult and not so well defined problems that provide rich opportunities for learning. Many Scratchers give up at these difficult roadblocks, however, those that find success, do so by asking for help from family, friends, or the community[10]. In the Scripts Workshop, we focus on creating a support network for Scratchers who have reached these roadblocks. They are connected with other Scratchers who work through the problem with them so that they can continue with their projects.

**Hard Fun**

Though it’s important that a learner enjoy what they’re doing, Papert points out that it does not mean it should be easy. Learners enter the Scripts Workshop when they are stuck or struggling with a problem on their Scratch project. The idea is they are motivated to work through it because they enjoy what they are working on. In the Scripts Workshop they find a community who can support them through that challenging process.

**You Can’t Get it Right Without Getting it Wrong**

Papert suggests the only way to ‘get it right’ is by carefully looking at what happened when it went wrong. The Scripts Workshop aims to connect learners with the people and resources to help them do that.
1.4 Social Perspectives on Learning

For much of history, teaching and learning happened within the context of apprenticeships. Though that’s less so the case today, in the learning sciences there continues to be much research to support the social aspects of learning. Vygotsky, one of the earliest proponents of the social aspects of learning, put forth the concept of the zone of proximal development (ZPD), which essentially accounts for what a child can do without help and what he or she can do with help. He defines it 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” [11]. Later, Lave & Wenger [12], studied communities of people that share a common craft or profession such as butchers and tailors. They coined the term ‘community of practice’ (CoP) to describe a model of learning whereby members of a community develop individually by sharing their knowledge and experiences. Newer members entering the community interact with more experienced members to learn the practices and tools of the community. More recently, Bruckman [13] studied young people in an online constructionist environment called MOOSE Crossing, where she found that kids looked to the community as a source of both emotional and technical support.
Design Inspirations

In this chapter I describe the design inspirations for this thesis. I start by describing the design of the Scratch Online Community where the Scripts Workshop is implemented and end with a description of the Stack Overflow programming community, the design of which also informs my thesis.

2.1 Scratch Online Community

Scratch is a popular block-based visual programming language that allows young people to create interactive media such as stories, animations, and games. The Scratch online community connects Scratchers (as Scratch users call themselves) allowing them to share, discuss, and remix one another's projects (Figure 3-1). There are similar commercial and research-based projects such as HopScotch and Alice[14] among many others. However, Scratch is unique in the size, diversity, and activity of its online community. There are more than 3 million registered users from all over the world, from ages 7 to 70, sharing 10,000 new projects a day. The online community was created to provide Scratchers with a community where they can support, collaborate, and critique one another, as well as build on one another's work. It aspires to appeal to people who may not have seen themselves as programmers and to be accessible for people of all ages, backgrounds, and interests [15].
Social Interactions to Support Learning

Within the Scratch online community there are multiple features designed to support social interactions for learning. These include the ability to view or remix the code of any shared project on the site as well as the ability to comment (Figure 3-2), on projects to leave feedback or constructive criticism.
Figure 2-2: An example of comments on a Scratch Project

In addition, Scratchers can use the backpack (Figure 3-3) feature to borrow pieces of code, or other assets from someone else’s project and use them in their own.

Figure 2-3: The backpack within the Scratch editor for moving assets between projects
There are also discussion forums for having conversations on a variety of topics and studios, which are collections of projects and comments curated by a group of Scratchers (Figure 3-4).

![Scratch discussion forums and studio example](image)

**Figure 2-4:** Scratch discussion forums (back) and an example of a Scratch studio (front).

**Culture**

The Scratch online community strives to be a friendly, welcoming, and productive space for beginners and experts of all ages, with diverse backgrounds, and a diversity of interests. This is a tall order for any online community and becomes more and more difficult as the community grows in size. The design of spaces and features within the community are heavily motivated by this goal. Members of the community are also expected to play a significant part in maintaining and supporting the culture. This is reflected in the Scratch community guidelines (Figure 3-5).
2.2 Stack Overflow

The Stack Overflow site is one of the more popular Q&A sites for programmers [16]. It is geared towards professional and enthusiast programmers looking for answers to their programming questions. Though this is not the target audience of the Scripts Workshop, Stack Overflow serves as a good example of the way a site can design its software and technology to shape the culture of its community. In describing the culture of Stack Overflow, co-founder Joel Spolsky [17], suggests that everything about a community either draws people in or turns them away. In examining the homepage of Stack Overflow (Figure 3-6) he suggests, “if you’re not a programmer, you see this page, you don’t understand a single thing and you leave, which is what we want”.

---

Figure 2-5: Scratch community guidelines
People often wonder why Stack Overflow is so strict about its protocol for asking and answering questions. Spolsky says they intentionally don’t support conversations. “If it’s fun, it shouldn’t be happening on our site” [17]. This is because their goal is not to support relationships or learning, their goal is to create reusable artifacts. He goes on to say that their main focus is not on the experience of users asking questions, or even the experience of users answering questions. Instead, they focus on the experience of users that come to Stack Overflow just to read answers or make use of previously asked and answered questions. By not allowing conversations, displaying an intimidating homepage, and closing questions because they are duplicates or poorly worded they support their goal of creating a set of re-usable artifacts.
In the Scripts Workshop, we attempt to build on the success of Stack Overflow’s community driven model but focus instead on supporting beginners and meaningful connections for learning first.
Existing Scratch Support Spaces

In this chapter I survey existing avenues of support within the Scratch community. As a participant in the Scratch community there are several spaces to get support while creating projects. This varies from spaces and content created and maintained by the Scratch team, to spaces designed by the Scratch team but mostly maintained by the community, to spaces and resources designed and maintained by the community. Figure 4-1 attempts to illustrate the current landscape of resources.
3.1 **Scratch Team Created Resources**

The Scratch team has produced and curated a set of tips, starter-projects, and tutorials that are meant to help in getting started. The tips exist within the programming editor and the tutorials and starter projects are organized in a central "Help" section of the site (Figure 4-2). They are created by learning researchers and designers, so these resources are well designed, appropriate, and useful; however, they are limited in number and scope. There are currently ten video
tutorials, twenty-four starter-projects and twelve how-to tutorials. These resources attempt to cover themes that are popular in the community such as animation, games, interactive art, music and dance, stories, and video sensing. Because of their visibility and usefulness these resources get a lot of use. They provide great starting points for project creation but it would be impossible for the Scratch team alone to support the breadth and depth of things Scratchers are interested in doing or get stuck on while they are doing those things.

Figure 3-2: Scratch help section (left) and in-editor tip-bar (right)

3.2 Scratch Team Designed, Community-Supported Spaces

In order to provide support beyond the finite resources the Scratch Team has developed, there are also explicit spaces designed for the community members to support one another.
Discussion Forums

The discussion forums allow Scratchers to connect on a variety of topics. The "Help With Scripts" forum welcomes Scratchers to post their requests for programming help and wait for a response from the community. Brennan[10] found there was a 95% response rate to 1449 topics (mostly representing questions) created from January to July 2012 with a median of 4 responses per topic. The forum is very active, with a median first response time of 22 minutes. However, this activity comes from a small (less than 10%) part of the community. One challenge with the forums that Scratchers reported in interviews with Brennan[10] was the ability to connect with someone who is able to help them in a way that is accessible.

Welcoming Committee

The welcoming committee is not a space for explicit programming support, but it is one of the first places many new Scratchers go and it reflects a space within the community that serves as a form of social support. Scratchers within the welcoming committee promote the cultural values of the Scratch community and are an initial connection for new Scratchers (Figure 4-3).

![Scratch Welcoming Committee](image)

**Figure 3-3:** Scratch Welcoming Committee
3.3 Community-Designed, Community-Supported Spaces

This is perhaps the most interesting of the existing support spaces. There are thousands of community-created projects tagged "tutorial", dozens of YouTube videos and several websites created by Scratchers wanting to support their peers and share what they know. They're created by all levels of Scratchers - from experienced to those who just joined a week or two ago.

Wiki

The Scratch wiki is a collaboratively written wiki written primarily by and for Scratchers. It consists of content curated and edited by approved contributors. It started as the Scratch Programming Wiki - a separate website created by a member of the community to help Scratchers with programming. As its popularity grew the Scratch Team saw its value and decided to host it on the main scratch.mit.edu site. Today the wiki includes more than 100 tutorials created by Scratchers. However, with titles like "Shooting projectiles" and "Extensive scrolling engine", these tutorials serve mostly those interested in creating games and use language that assumes a certain level of technical experience. For those who need more guidance, the format of the wiki doesn't provide an opportunity to interact with the author or ask questions.

Helper Projects and Studios

The thousands of projects tagged “tutorial” represent a range of interests - from drawing, to animation, to games, and vary from general "how to do variables" to very specific "how to draw a pony facing you" (Figure 4-4). They also vary greatly in terms of their usefulness and understandability. Further, because they aren’t highlighted anywhere on the site many of them are never discovered and have only a few views from the community.
Scratchers have also used studios to create spaces meant for helping fellow Scratchers. An example of this is "Tutorial Madness" (Figure 4-5) which is managed by fifteen Scratchers and contains more than 100 tutorials.

Figure 3-5: Scratchers created community of tutorials and helpers
A look at the comments, more than 1000, with the majority being requests to contribute, reflects the enthusiasm felt about helping others in the community (Figure 4-6). The enthusiasm and diversity of activity within these spaces is inspiring however it's unclear how useful they are to other Scratchers. They aren't organized or easily found within the community - and the quality varies drastically.

Figure 3-6: Comments from Scratchers wanting to contribute to the Tutorial Madness studio

### 3.4 Limitations

Many kids thrive in the Scratch online community and are able to find resources that provide
support and inspiration as they create and share projects. In interviews with Scratchers conducted by Brennan [10] most kids described the ability to connect with other creators in the online community as an important and useful resource towards their own participation as creators. However, we also know that many Scratchers don't ever make those connections and don't get past the creation of a single project. Since these Scratchers are inactive or end up leaving the community, it's difficult to tell their story or understand their feeling of disconnectedness.

Even active members of the Scratch community run into challenges finding the resources they need in the community. The size of the community is both a challenge and an advantage in this regard. With a larger community there are more potential sources of support, however, it is also significantly harder to find the right sources in the crowd. Another challenge is communication - many Scratchers are unable to articulate the problem and support they need[10].
Scripts Workshop

In this chapter I introduce the Scripts Workshop, an experimental learning environment that connects the people, spaces, and activities required for effective community-driven support within the Scratch online community. The design of the workshop is based on principals of social and constructionist learning as described in Chapter 2 and influenced and informed by existing Scratch community practices described in Chapter 3. The Scripts Workshop is developed on top of existing features in the Scratch community and actively avoids implementing new platform structures. In product development this is sometimes referred to as a minimum viable product (MVP) strategy. It allowed us to test and iterate our design with minimal resources. The following sections provide a description and analysis of the main spaces, people, and activities within the Scripts Workshop experiment. Section 5.4 concludes with 4 community design principals that can be derived from this experiment as well as an outline of its limitations.

4.1 People

The people involved in the Scripts Workshop can be divided into 4 major roles: Learners, Script Helpers, Scratch Mentors, and Scratch Team. In the workshop, learners connect with Script Helpers and Scratch Mentors who are supported by the Scratch Team. These roles are derived from Illich’s requirement that learners have access to a network of skill models (Script Helpers)
and pedagogues (Scratch Mentors) that is supported by administrators (Scratch Team). The following sections examine these roles within the context of the workshop and each other.

**Learners**

Many of the community-designed and community-supported spaces in Scratch have a surplus of Scratchers wanting to help and a deficit of Scratchers looking for help. For a short time, the Scripts Workshop was featured prominently on the homepage of Scratch in order to direct more Scratchers seeking help to the space by reducing the current barriers to finding these community support spaces. Over the course of 3 months more than 100 projects went through the workshop, 30 of them were from Scratchers who had only been in the community for a few months and several of them included first projects. In the Scripts Workshop the number of people seeking help often outnumbered those providing help, which gave those providing help ample opportunity to interact with real learners and real problems, something that is missing in other Scratch community support spaces (Figure 5-1). In the discussion forums Scratchers often request help without context and the helper studios tend to be spaces occupied by those wanting to help (Figure 5-2).

![Scratch Comment](http://scratch.mit.edu/projects/25127031) trying to add a scrolling map. and a save system can anyone help?  
31 July 2014  Reply

![Scratch Comment](http://scratch.mit.edu/projects/24401011) i'm trying to turn this into a scroll for ages. can anyone please help me?  
31 July 2014  Reply

![Scratch Comment](http://scratch.mit.edu/projects/25165670) Could someone please have a look at this?  
30 July 2014  Reply

**Figure 4-1:** Comments within the Scripts Workshop reflect people looking to get help
Figure 4-2: Comments within a typical “Helper Studio” reflect people looking to provide help

**Script Helpers**

The Script Helpers were selected by application. There is an abundance of enthusiasm around helping that doesn't correlate with ability to help, and the application process introduced a necessary threshold to participation that was meant to filter those with at least some experience and demonstrated ability to help. The application involved creating and submitting a tutorial project that Scratch Mentors (described below) would review and either accept or provide feedback on for improvements. The tutorial project wasn't necessarily a good indicator of someone's ability as a Script Helper; rather, it was an assurance that they had some minimal level of experience as a creator in Scratch and an invested desire in sharing that skill.

It's important to note that the Script Helpers didn't have any special privilege in their role as helpers. That is, in keeping with Illich's [6] design, where anyone with a skill to share may share it, the Scripts Workshop was open to everyone. If a Scratcher saw a project in the workshop that they felt they could provide help on, they were welcome to do so whether they were a Script Helper or not. Instead of privilege over helping, the Script Helpers instead had a responsibility
for helping. They were visible, active, and skilled, members of the community who participated in the space. They also received explicit support from Scratch Mentors in developing their ability as helpers. Script Helpers' interactions within the Scripts Workshop act as models for other Scratchers entering the space, with the desire to help but lack of experience and ability.

Over the course of three months, 46 Script Helpers were accepted into the Scripts Workshop. Most of these were Scratchers who had some experience and skill expertise in Scratch, were excited about supporting others, but lacked experience in how best to do so. In fact, many of them felt the goal of providing support was to do so as fast as possible which sometimes meant just solving the problem without providing any explanation (Figure 5-4).

![Image of conversation]

**Figure 4-3:** Script Helpers discussing importance of quick help

**Scratch Mentors**

While the Script Helpers focused on helping Scratchers with their projects, the Scratch Mentors focused on helping Script Helpers better help Scratchers. They did this by modeling good strategies for help and also acting as moderators for the Scripts Workshop spaces. Scratch Mentors, play a role akin to what Illich [6] describes as 'Professional Educators', and are focused on supporting the learning experiences of others. Though Illich imagined this role filled by
people with professional experience as educators, we explored the capacity of young people to take on these responsibilities. The Scratch Team invited 5 older teens to serve as Scratch Mentors, all of whom were selected based on their active participation in the community, specifically, helping others with a demonstrated empathy for beginners.

For example, all 5 Scratch Mentors were also part of the Scratch Welcoming Committee described in Chapter 3, so they had experience interacting with newcomers and beginners and promoting a supportive and respectful culture. They were seasoned members of the community who contributed and participated in multiple spaces from the wiki to the discussion forums and other social spaces.

In the Scratch forums or wiki Scratchers who answer forum posts or create wiki articles don’t have much explicit support in these roles. In the Scripts Workshop, Scratch Mentors model strategies for supporting Scratchers, provide Script Helpers with feedback on critique as they help Scratchers, and create resources and facilitate conversations that reflect on what it means to support someone’s learning.

Below is an example of a poor interaction between a Script Helper and a learner in the Scripts Workshop:
Figure 4-4: Unsupportive interaction between a Script Helper and Scratcher

This particular Script Helper had several interactions like this. Below is an intervention from a Scratch Mentor:
Figure 4-5: Scratch Mentor modeling supportive help

Scratch Team

There were 5 of us from the Scratch Team who led the design of the Scripts Workshop. The role we played would most closely match what Illich [5] referred to as the "administrators and designers" of learning webs, whose skills and attitudes he compares to that of museum or library staff. Their main goal is to stay out of the way while facilitating encounters among students and the things and people that will support their learning.

The 5 of us on this team, together, represent a skill set that encompasses UX design, web-development, programming, learning research, education, community moderation, and perhaps most importantly, we are all active members of the Scratch online community. Having some expertise in all these areas was important to the design of theScripts Workshop. We met weekly
to discuss the design of the space and (after launch) to discuss how it was being used, what was working or not working, and what we should change.

When the Scripts Workshop first launched we participated in order to help model the type of role we expected the Scratch Mentors to play. We also invited the Scratch Mentors to participate in some of the planning of the Scripts Workshop. We hoped that their participation in the design and plan for the Scripts Workshop would allow them to better understand and support the goals of the space, as well as empower them as managers of the space. After the initial launch we reduced our participation to a minimum, only making ourselves available to Scratch Mentors if they wanted our support in a specific decision or interaction.

4.2 Activities

There are two main activities that are supported through the Scripts Workshop: creating resources and providing help on projects. These are adapted from activities that already occur in the Scratch community. This section describes those two activities and their goals.

Community-Supported Creation of Resources

Within the Scratch community, Scratchers produce text-based tutorials in the wiki and project-based tutorials that they share on the site. In the Scripts Workshop we build on the idea of using Scratch projects as a medium for resources. They are not an ideal medium by any means. For example, showing the Scratch UI or blocks in a project (which tutorials often do) requires taking screenshots with third party software a process that is challenging for many young Scratchers and tedious in general. However, we feel that projects are a better medium for resources than text because they are interactive and more accessible for both helpers and learners. In the Scripts Workshop we attempt to provide Scratchers support in creating more useful and accessible resources.
As described above, Script Helpers apply to participate in the Scripts Workshop through the creation and submission of a "helpful project". We defined a helpful project as anything that explains a concept in Scratch or contains a set of reusable scripts or sprites with instructions on how to use them. Scratchers submitted projects and Scratch Mentors reviewed them and provided feedback and suggestions for improvement. Once they got to a place where a Scratch Mentor felt it was something that would be helpful, easily understandable, and usable by another Scratcher, the project was added to a growing collection of "helper projects" which served as models for others who wanted to apply. By receiving design constraints and goals from Scratch Mentors, Scratchers produced projects that were more relevant and useful for programming in Scratch than what you might see by just typing "programming tutorial" in the Scratch search bar. With access to a set of example projects in the Scripts Workshop, Scratchers had a model to work from and were able to borrow concepts and ideas in their own projects. Through feedback and suggestions from Scratch Mentors, Scratchers were able to refine their projects (Figure 5-7).

Through this process we ended up with a set of projects that ranged in topic from variables, dragging, sensing, lists, and cloning among others; as well as a set of projects with reusable scripts and sprites. Some projects even developed creative techniques that prompted the learner to participate. Figure 5-8 is an example of a project, which provides the viewer with two reusable animated sprites, and a third image, which they invite the viewer to program herself.
This is very very helpful and you explained the script very well! Just one little suggestion: Could you describe your usable script in the project notes? That would be even more helpful I think.
30 April 2014  Reply

Done!
30 April 2014  Reply

Awesome! I could already add the project to the Scripts Studio, but I thought maybe you would like it to look a bit more interesting for people who look at the studio gallery - so maybe you'd like to make a background or something like that first?
01 May 2014  Reply

OK!
01 May 2014  Reply

Figure 4-6: Scratch Mentor providing feedback on how to improve a tutorial project
The studio where these projects lived was highlighted on the site, which meant the projects received many more views than the average project labeled "tutorial" on the site. In addition, many of these projects received positive and thankful comments from viewers. Just as recognition from the community leads to confidence and motivation as a creator, we hope that it does the same here for helpers (Figure 5-9).
**Figure 4-8:** Positive remarks from community on a Scripts Workshop tutorial project

**Community-Supported Help on Projects**

In the Scratch Community, the discussion forums are one space where Scratchers go to get help from another Scratcher. However the forums are text heavy, and questions are often asked without context, and responses often use language that's not understandable to the asker, leading to confusing conversations that are usually left hanging (Figure 5-10).
Figure 4-9: Scratcher looking for help is left hanging in the discussion forums

There are of course other resources available in the Scratch community (as described in Chapter 3); however, for a newcomer in the community (or even an old-timer) they aren't so easy to navigate and understand.
In the Scripts Workshop we shift the focus to helping Scratchers with a specific problem within the context of a project they're working on. The workshop is an entry point for getting help on a problem. Within the workshop a Scratcher connects with a Script Helper, who will either help them directly or will act as a guide for the vast ecosystem of resources that exist in Scratch, by connecting them to other relevant sources like tutorials, example projects, or other Scratchers. (Figure 5-11)

![Image of forum conversation]

**Figure 4-10:** In providing support a Script Helper points to another Scratcher who may be useful

Scratch Mentors also participate in helping projects in the Scripts Workshop - the goal is for them to model behavior of what it looks like to be empathetic towards learners and provide good help. In addition they provide Script Helpers with feedback on their interactions with Scratchers in order to help them refine their techniques for helping (Figure 12, 13).
Hey guys, I know that in this, you're supposed to make a project with your question, but right now my 'project' is no more than an idea. So if you don't answer this, it's fine. But my question is: I want to make a game where things fall from the sky and you must catch them. Simple. But I don't know how to randomize the amount of time in between the things falling. If you could show me a script or something, that would be epic, but if you don't that's completely fine.

01 May 2014  Reply

Not a problem! Instead of using wait 1 secs, put the pick random 1 to 10 from the operators section into the number 1 in the wait block. Clear?

01 May 2014  Reply

Oh... I thought you are supposed to see the project before saying it...

01 May 2014  Reply

Well...the point of this studio is for people who have projects they need help with, so I'm not really sure if this is the right place for this. Maybe you could post this in the "Help with Scripts" section of the forums.

01 May 2014  Reply

Figure 4-11: Script Helpers responding poorly to a request for help.
4.3 Spaces

There are three main spaces that define the Scripts Workshop. These are: the workshop – a communal space for connecting, project comments – a space for contextual conversations, and helper forums – a space for reflection. The following sections describe and analyze these three spaces.

Workshop Studio

The workshop studio is a Scratch Studio that highlights projects looking for help and people who are able to help. Unlike many of the existing support spaces within Scratch, upon entering you immediately see that it's a space for beginners or people seeking help and not just a place for experts (Figure 5-14).
**Figure 4-13:** Main Scripts Workshop space with listing of projects looking for help

You can also easily identify people in the space who can help on the curators tab (Figure 5-15, 5-16).

**Figure 4-14:** Scratch Team and Scratch Mentors participating in the Scripts Workshop, as listed in the ‘curators’ tab of the Scripts Workshop
Unlike other spaces, which are text heavy, the workshop uses the familiar and thus more welcoming elements of projects and people to define the space. The comments tab is where Scratchers can submit a link to their projects and request help. There aren't long or confusing conversations here, only short interactions asking for help which makes the space less intimidating and lowers the barrier for participation. The communication within this space is also moderated by the Scratch Mentors who promote and model patience and empathy.

The design of the workshop studio acts as a signal to beginners and learners that it is a space for them and provides clear and easy examples for how they can participate. In addition, the workshop space acts as a model for all those Scratchers with an enthusiasm for helping who want to develop their skill in doing so. As described in the "Script Helpers" section above, the workshop space is open to anyone who wants to provide support whether they are a Script Helper or not. Thus, these enthusiastic aspirational helpers can go from observing, to practicing
participation within this space and receive feedback from Scratch Mentors and others as they do it (Figure 5-17, 5-18).

Figure 4-16: A Scratcher looking for help in the Scripts Workshop

Figure 4-17: The same Scratcher from figure 5:17, now providing help in the Scripts Workshop.
The workshop space acts as a starting point for those seeking help, and the people within the workshop act as knowledgeable guides to the world of Scratch resources that is otherwise overwhelming or un-navigable by a beginner.

**Project Notes & Comments**

Many of the conversations in the discussion forums are out of context and confusing to follow and often lead to frustration by participants. In many cases it seems a conversation will just drop off and it is unclear whether the conversation starter's issue was resolved or whether they just decided not to return to that thread in the forum. In the Scripts Workshop, a Scratcher frames their request for help in the project notes and help is provided in the project comments. This places the 'object' the learner and helper are referring to "right in front of them", making it easier to talk about. It also makes it significantly easier to start, follow-up, or end a conversation around a project. Projects in the Scripts Workshop aren't removed from the workshop until the learner confirms that they don't need anymore help (Figure 5-19).
Hello! I am on the Scripts Team. I think your problem is that on the bottom of your define "Set Basics" script you have a block that's waiting until Eldertalk = 3. All you need to do is remove that block!

08 August 2014  Reply

Alright. Let me try that.

09 August 2014  Reply

YAY IT WORKED THANK YOU!!

09 August 2014  Reply

No problem! Is that all you needed help with?

09 August 2014  Reply

Yup! Or you can test it.

09 August 2014  Reply

I did! It works great! :) Since you don't need any more help, I'll have it removed from the Script Workshop.

09 August 2014  Reply

Figure 4-18: Example of a Script Helper proving help on a Scratchers project
Helper and Mentor Forums

A key space within the Scripts Workshop is the forums where Scratch Mentors and Script Helpers share stories, strategies, and questions among themselves. These forums provide a space for Scratch Mentors and Script Helpers to reflect on the process of helping and support one another in that process. They also use this space to help organize themselves in order to better work as a team. Having a space where they can communicate helped to encourage a sense of community among them.

In the Scratch Mentor forums one Scratch Mentor shared her thoughts on how someone should be helped, drawing on her experiences as a "homework helper" outside of Scratch:

*How should they be helped?*

I've been homework mentor in my school… Our strategy was the following (which worked most of the time):

* Get clear about the instructions, what do you want to achieve? (most of the time, people didn't understand the instructions because they didn't read them correctly)
* And then: How will you achieve it? I'm sure you have some ideas, collect them. (Makes them think about solutions, not only waiting for someone to answer)
* Then think about them again, what can you exclude? (Kind of scientific way of doing it But it's really cool to reflect about one's own ideas)
* If you really don't know the answer, ask me again (so we think about the ideas together and find the right solution)

*Figure 4-19: Advice from a Scratch Mentor on how to provide help*

She also shared her insights into the challenges faced when helping. For example, the technical challenges presented by working online:

*But from my experiences I can say, that it's not that easy on the internet - you are far away from the one you are talking to, and sometimes you aren't even online at the same time. Also, it's hard to describe programming and it's easier to provide the scripts.*

*Figure 4-20: Scratch Mentor describing challenges of online support*
Or the challenges of helping with programming:

Most of the time I give some ideas, like: “You could use this and that block” or “You need a script for this, that and that” (so that they have to figure out themselves how to put the script together) or else I link a wiki-article, if there’s a good one...

But that's not so good for long and complicated scripts: What I do then is make the scripts and add comments to them and then I try to reflect with the user what I did, but I can imagine that some people just say “Yes I understood that” although they didn't understand it or just didn't want to think about it.

**Figure 4-21:** Scratch Mentor describing challenges of programming support

Or managing learner expectations without discouraging them:

Another experience: Sometimes, people are asking: How do I make a multiplayer game like Minecraft. Often they are new Scratchers without any projects. How do I explain to them that they'll probably not be able to do that in the way they imagine it, without discouraging them? Most of the time I’ll link a wiki article or a project here, just for them to get an idea of what they plan.

**Figure 4-22:** Scratch Mentor describing challenges of pedagogical support

This led to discussion among mentors about how to convey these ideas and the suggestion came up to create a Scratch Helper Guidebook. After more discussion this turned into a collaborative Scratch project made by the Scratch Mentors to convey what it means to be a "helper" (Figure 5-24).
In the Script Helpers forums one Script Helper shared his story of helping:

*I helped helloyowuzzup on their project Scratch's Singathon. They basically had everything ready, but they hadn't done much programming. They needed help with the menu to flip through the different songs. In the project notes they said that their own scripting was confusing them, so I decided it would be best if I just made a remix for them with a functional menu.*

*Before I started on the remix, however, I made sure I understood exactly what they wanted. I had a fairly lengthy back-and-forth to make sure I did it right. During this back-and-forth, they said that they didn't understand something that I had said. After they said so, I did my best to explain things more simply, and that way I was able to find...*
out exactly what they wanted most efficiently.

When I started remixing, I found out the project was horribly cluttered; not with scripts, but just in general. There were a lot of sprites, and they were all showing in random places. The scripts that were there were very repetitive. Accordingly, I started with just some simple organization, and then I went on to putting together the menu. That part wasn’t hard. After I finished it, they were VERY thankful, which made me feel good about helping them.

He goes on to summarize the take-away from his experience as:

It’s not just about fixing the problem; it’s also about making sure they have a good experience and get exactly what they want, and, as (in most cases) more experienced Scratchers, showing them what Scratch is all about. And, hopefully, the Scratcher will be very thankful, making it enjoyable for you as well!

The Script Helpers forum was also used as a space self-organize. One Script Helper suggested that it would be easier if he knew what skills all the other Script Helpers could help with so he could more easily point to them as a resource. He decided to make a chart and invited all the Script Helpers to fill it out (Figure 25).
Figure 4-24: A chart of individual Script Helpers areas of expertise. Names have been greyed out.

4.4 Possibilities and Limitations

The Scripts Workshop illustrates the possibilities for more effective community support for learning within the Scratch online community. Specifically, we can highlight 4 major takeaways:

1. Focusing on supporting projects instead of questions helped make otherwise abstract conversations more concrete.
2. By creating spaces and highlighting people that model good strategies and techniques for helping, the Scripts Workshop provides Scratchers who want to help, with an opportunity to observe and then participate.
3. Defining and supporting roles focused on mentorship and pedagogy helped to seed and
promote a culture focused on learning and not just answering questions.

4. Creating spaces for reflection also strengthened the focus on learning and provided opportunities for helpers to support one another as they develop and share strategies and stories.

The Scripts Workshop also highlights some limitations:

**Identity**

Scratchers struggled with the idea that Script Helpers provided help, but so could anyone else. The fact that this was a distinct role, that required an application, review, and acceptance process, signaled to them that there should be some unique privileges that came along with it. This conflicted with the general idea of promoting helping as practice and value embedded in the community.

**Communication**

Another limitation is introduced by the block-based nature of the Scratch programming language. Script Helpers and learners would struggle through conversations where they tried to use text to explain how to edit, shift, add, or remove graphical blocks. Sometimes they would remix and share a project to overcome this. These projects would clutter their own profile pages and so they'd delete them soon after and the learner would lose access to them.

**Diversity of Interest**

Many of the projects submitted to the Scripts Workshop were by newcomers to the Scratch community. However they lacked diversity of interest. The majority of projects submitted were games. Though this could be because Scratchers are more likely to run into scripting issues when creating a game than other types of projects like animations or interactive stories - I'm skeptical of that. A community support space should strive to better reflect the diversity of interests of the
community.

Some aspects of these limitations can be addressed through more substantial platform changes or additions. I'll propose and describe these in Chapter 6.
Implications: Scratch Community

The design of an online community greatly influences its practices. Design decisions within the Scratch online community are made with careful consideration as to the implications it will have on its core values of creating and sharing in a supportive and diverse community. In this chapter, based on the learnings from the Scripts Workshop, I suggest and describe design changes within the community to encourage and better scaffold Scratchers supporting one another. I'll structure these suggestions according to three key limitations we saw within the Scripts Workshop: Identity, Communication, and Culture.

5.1 Identity

Within the Scratch Community's 3 million plus registered users, it's difficult to identify who might be willing and able to provide help. The Scripts Workshop defined a space where Scratchers who were willing to help congregated; however, their ability to help varied. Since many young Scratchers are excited to provide help, we had a large number of applications to become Script Helpers but very few that qualified. In a later iteration we moved to an invite model based on previously demonstrated ability and interest in supporting others. This would work well in the limited scope and size of the Scripts Workshop but doesn't scale well to support a community as large as the Scratch Community. To do this, we need better ways of identifying Scratchers who have both the ability and willingness to help. We also want to ensure that helping
doesn't become an activity that is reserved for a selected few.

Currently Scratch profiles focus on identity in the community as creators. For example, the content on a Scratcher's profile page lists their shared projects and studios, favorite projects, and followers (Figure 5-1). Many Scratchers who participate in the community as helpers and create tutorial projects or remix projects as a form of helping, create separate accounts to draw a more clear line between their identity as creators and helpers. If instead, their profile also listed tutorials or resources they've created, projects they've helped troubleshoot, and skills they are willing to share (Figure 6-1) - it would better highlight not just how they participate in the community as a creator but also as a helper or skill model. It would also provide Scratchers with the willingness to help, clearer pathways to build up and demonstrate their ability to help.

Figure 5-1: Scratch user profile page
If stored and indexed explicitly this data could also be used to move towards a stronger network-of-skills model allowing Scratchers to better find helpers within the community – for example, being able to search based on skills or tutorials, or highlighting the most loved, favorited, or viewed tutorials on the homepage.

By tying these forms of participation more closely with a user's profile, it will also increase the perceived importance of them and foster a community that values not just creating but also supporting others in creating.

### 5.2 Communication

In the Scripts Workshop we saw Scratchers struggle with communicating their ideas while helping others on a project. Part of this is developmental and can be attributed to the young age of many Scratchers, but part of it is also due to a lack of appropriate tools that support communicating about projects.

In the Scripts Workshop, Script Helpers helped debug scripts through conversations in the
project comments. In contrast to the discussion forums, this helped to better contextualize the conversations around specific projects. Still, Scratchers continued to struggle with the abstraction required to talk about scripts using just text-based comments. The block-based design of the Scratch programming language makes it more accessible to beginners, but it also adds complexity when trying to discuss the scripts they construct. In the physical world, kids collaborating or helping each other on a project can more easily communicate by using non-verbal cues like pointing to specific areas or “showing” instead of “explaining”. To facilitate conversations about projects in the online space Scratchers need better tools for talking about them. In the discussion forums, Scratchers are able to assemble graphical scripts (albeit with some difficulty) in their posts. Through the Backpack (Figure 6-6) Scratchers can collect and share sprites, scripts, costumes, and sounds between projects.

![Backpack](image)

**Figure 5-3:** Backpack for moving sprites, scripts, sounds, and costumes between projects.

Since its release with Scratch 2.0 in 2013 the Backpack has made it easier to build off the work of others in the community by borrowing bits and pieces of a project instead of remixing the entire thing. Focusing on supporting one another, the Backpack could be extended to allow Scratchers to drag scripts out of the backpack and drop them into a comment, fully integrated with the text. This would help them to talk about and debug bits and pieces of a project instead of remixing the entire project. This would give Scratchers a richer and more concrete language for talking about projects (Figure 6-7).
5.3 Diversity of Interest

Scratch has demonstrated the value of interest-based approaches to computer programming and computational fluency[18]. A core element of the Scratch programming language and community is reflecting a diversity of interests as a way of engaging people who may not typically be interested in programming. The homepage (http://scratch.mit.edu) of the Scratch online community is actively curated to feature projects that highlight a range of interests and
styles, which include dress-up projects, music projects, political surveys, and animations among others. In designing support spaces for this community it is important then that these spaces also reflect and support the diversity of interests of the community. This is an area where both the existing community support spaces and the Scripts Workshop fail. As a next iteration of the Scripts Workshop we are looking at creating sub-spaces for support based on interest. We can think about using these spaces to better support groups who are traditionally underrepresented in computing by connecting to their interests [18] and providing a context with strong social support and guidance. By creating multiple spaces for support based on interest we can also focus on smaller sub-communities at a scale where it would be easier to define culture and apply features of the Scripts Workshop.
Implications: Beyond Scratch

In this chapter I will describe implications of this work beyond Scratch as well as future research possibilities.

6.1 Relationships and Inspiration

An important aspect of learning is relationships. When reflecting on childhood learning experiences people often refer fondly to the people who inspired them, whether it be teachers, friends, or family members. In Illich’s [6] learning webs the interactions are framed as exchanges with not much indication of building longer lasting relationships. However, relationships are an inseparable part of communities, and relationships are powerful tools for learning. In the physical world these connections are often unplanned or serendipitous. In the design of online learning webs we should consider how relationships fit or don’t fit and what we can do to support connections between learners and people who will not only share their skills but be a source of inspiration and more personal mentorship.

6.2 Technologies for communication

The way we connect and communicate over the Internet is changing rapidly. Today there are
tools like Skype and Google Hangouts to communicate in face-to-face in real-time using video over the network. Wearable and mobile technologies also provide new ways to think about how and when we connect with one another. In this thesis learners communicated through asynchronous text-based methods; however, there are opportunities to explore how we can utilize these new technologies to improve connections and communication for the purpose of learning. It would be a mistake to assume that these newer technologies are always a better choice over older ones. For example, real-time video communication is appealing because it’s closer to the way we communicate in the physical world, but it also requires planning and coordination. There are also considerations to be made about the most effective methods of online communication for kids. As we design and develop new online learning environments it is important to explore and understand the challenges and opportunities for communicating virtually and what types of communication are best for what types of learning interactions.

6.3 Policy

As we think about designing online learning environments for young learners we inevitably have to revisit policies around kids interacting online. At the moment many of the policies, though meant to protect children from the perils of the Internet, also exclude them from taking advantage of its potential as a powerful tool for connecting and learning. Today, most online learning environments focus on adult learners; however, reform in policy could lead to a burst of new environments that support kids learning online.

6.4 Massive Open Online Courses

The increasing interest in teaching and learning online goes beyond the field of computer science. As we think about learning in the online space we have again opportunities to reconsider our instruction-focused methods of pedagogy that for years have remained unchanged.
One of the most prevalent systems of online learning today comes in the form of Massive Open Online Courses (MOOCs). There are MOOCs offered in subjects that include computer science, engineering, business, education, and more [19]. MOOCs are divided into three categories: network-based (also referred to as cMOOCs or connectivist MOOCs), task-based, and content-based (like Coursera and Udacity, also referred to as xMOOCs and the most popular of the three) [20]. Though most applications combine aspects of all three of these categories they generally have more properties of one. xMOOCs more closely imitate learning in schools and the goal is often to reach broader and larger audiences by making content available online. cMOOCs are much less common but focus on the networked aspects of both the Internet and learning. The fact that there are relatively fewer cMOOCs may be in part because of the uncertainty of how to design them well to support social interactions of potentially large groups of people or strangers. MOOCs have only been around for a few years, but as they continue to evolve, it is important to research ways of better combining the positive learning aspects of all three types while addressing the question of who and what learners should be connected to (and how) for a different subject areas.

6.5 Internet as Learning Web

The Internet itself is essentially a giant network of people and things – many of which are useful for learning. Sites like Facebook act as a “social-layer” to the web – connecting friends, family, and colleagues for sharing stories, photos, and other personally meaningful artifacts. It’s possible to imagine a “learning-layer” to the web – connecting mentors, skilled-experts, and peers for joint-inquiry, skills exchange, and sharing information and resources along various subjects and areas of interest. Google Helpouts is a new video-based platform and marketplace for skill sharing that takes a step in this direction. An ideal learning-layer would also provide opportunities to connect people based on location in order to support communities of learners both offline and online, and would support learners of all ages.
Bibliography


