# Tinkering with ScratchBit:
# Explorations in Blended Making

by
Kreg Hanning

Bachelors of Science in Liberal Studies, Specialization in Technology in Education
Lesley University, 2016

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 2018

Signature redacted

Signature of Author ________________________________
Program in Media Arts and Sciences
August ~~24~~, 2018

Signature redacted

Certified by ________________________________
Mitchel Resnick
LEGO Papert Professor of Learning Research
Program in Media Arts and Sciences, MIT
Thesis Supervisor

Signature redacted

Accepted by ________________________________
Tod Machover
Academic Head
Program in Media Arts and Sciences

# Tinkering with ScratchBit:
# Explorations in Blended Making

by Kreg Hanning

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

## Abstract

In recent years, maker and coding movements have gained significant traction in learning communities around the world. To meet the needs of these movements, various forms of physical and digital construction kits have begun to emerge. Often times these construction kits facilitate creation in only one domain, either in the physical or virtual world, but not both. For my Master's thesis, I propose a new system, called the ScratchBit, that attempts to merge both physical and virtual making into a single cohesive experience. I am coining a new term, *blended making*, to describe the style of making that this system will enable.

Blended making is the process of engaging in construction in both the physical and virtual world -- and with interplay between the two. Some examples could be designing a physical costume that allows the wearer to become an actor in a digital story, or creating a set of custom LEGO handlebars to control a flying bicycle video game on the screen.

The ScratchBit is designed to enable young learners to engage in blended making. With an emphasis on composability, the ScratchBit allows almost any material -- such as cardboard, dolls, sneakers, or even swing sets - to be transformed into a physical interface for projects created with the Scratch visual programming language.

This thesis presents analyses of projects that children created using the ScratchBit and discusses how these analyses influenced the iterative design of the ScratchBit. In addition to documenting and commenting on the iterative design process, this thesis also presents

classifications of the types of ScratchBit projects that children created and guidelines for designing systems that support blended making.

# Tinkering with ScratchBit: Explorations in Blended Making

by

Kreg Hanning

Advisor ________________________________ Signature redacted

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

# Tinkering with ScratchBit:
# Explorations in Blended Making

by

Kreg Hanning

Reader

Signature redacted

---

Karen Brennan
Associate Professor of Education
Harvard Graduate School of Education

# Tinkering with ScratchBit:
# Explorations in Blended Making

by

Kreg Hanning

Signature redacted

Reader

Brian Silverman
Playful Invention Company
Co-founder and President

# Contents

# List of Figures

# 1. Introduction

## Making Makes Sense

Growing up in the US public education system, I often struggled academically. It wasn't necessarily because I had difficulty understanding a particular topic or subject. Often, I just wasn't motivated. I found it difficult to learn about things that felt entirely disconnected from my own life. Each school day seemed as predictable as the last: moving from class to class, from subject to subject, at the direction of the bell.

Despite struggling through many of my courses, I usually excelled in courses that involved computers or technology. Naturally, I thought that I must have some talent in technology and just not the rest of school. It wasn't until the second semester of my Computer Science undergraduate program that I realized this was not the case. The university courses were usually taught in a very hands-off way. Instead of working on projects, I would sit through long lectures on programming languages or take handwritten exams on data structures. I realized I was learning more through my own side projects outside of school than I was through my classes. It's because the projects I took on outside of school were personally meaningful to me. I was creating and exploring things that I was interested in.

After having this revelation, I spent some time reflecting on my past experiences in school. I realized that my success had largely depended on how much autonomy I was given. In high school, I can remember falling in love with a class on Wall Street investing. Anyone who knows me probably knows that I have little to no interest in financial endeavours. But for some reason this class still resonated with me a decade later. I realized that it was because I was given the freedom to control how I learned in this class. The semester-long assignment was to create a virtual investment portfolio with a final report due at the end. We were told we could invest into anything we wanted as long as we had good reason to do so. For example, being a Boston Red Sox fan was not a valid reason for investing in Kayem, maker of the famous Fenway Frank hot dog. I ended up using this assignment as an opportunity to explore and learn about topics that I personally cared about. Which at the time was most likely cars and sports. Even though I didn't have an interest in finance, I was willing to spend long hours researching businesses and reading articles to try to build my portfolio. I'm still not a financial guru, but to this day I remember the sense of pride I felt in building and sharing my portfolio throughout the class.

Whether I'm working on a virtual investment portfolio, hacking custom assets into a favorite video game, or learning to cook something new, I learn best when I am making something. For me making makes sense.

This idea of learning through making has led me to my current work which is to design tools for children to engage in the making process. This thesis will document two years of development of a new maker platform designed to allow children to create seamlessly in both the physical and virtual world.

## Children as Makers

All around the world, young learners are engaging in the act of making. Dale Dougherty, founder of Make Magazine, argues that we are all inherently makers. Referring to the term "Maker", Dougherty states that it "describes each one of us, no matter how we live our lives or what our goals might be. We are all makers: as cooks preparing food for our families, as gardeners, as knitters." (Dougherty, 2012). The examples provided by Dougherty are somewhat rooted in the adult world, cooking for our families and gardening; however, one doesn't have to think too hard to realize that young people too are deserving of this title, "Makers".

Children come up with stories and act them out using a wide range of makeshift materials. New games or sports are often invented on the playground, sometimes necessitated by a lack of proper space or equipment. Cardboard boxes are transformed to become race cars or boats for sailing the seven seas in search of lost treasure.

Making is often thought of with very concrete outcomes. Through making you produce a meal, a new machine, or an article of clothing. While this is true, I think it's worth highlighting that the process of making can be just as important as the final product.

Children's plays don't often end up on Broadway and their playground games aren't likely to become an event at the next Olympics. However, the process of making can provide a new way for children to explore and make sense of the world around them. It has the potential to shift their mindset so that they can see themselves as inventors, or creators, instead of passive participants in a world that has been predetermined for them. They learning by "constructing knowledge through the act of making something shareable" (Martinez and Stager, 2016).

Fortunately, we live in a time where tools for making are in no short supply, even for children. Craft materials, LEGO bricks, recyclables, and vinyl cutters are all examples of tools that children can use to bring their ideas to life in the physical world.

Tools for making digital artifacts are also in great supply. Using various software tools, children can direct their own movie productions, publish their own e-books or comic strips, and design websites and software of their own. Scratch (Resnick et al., 2009) is one example of a virtual construction kit that provides a way for children to design their own stories, games, animations, and a whole lot more.

For my thesis, I decided to think about the merging of these two styles of making for children, physical and virtual. I have come up with a new term to describe making across both domains, which I will refer to as *blended making*.

## Blended Making

Blended making describes the process of creating in both the physical and virtual world --- and with interplay between the two.

Blended making is not a novel idea. Quite a few systems currently exist which allow people to engage in creativity across both physical and virtual spaces. By using the term Blended Making, I am hoping to bring clarity to discussions around child-centered making and continue to challenge the perception of what young learners are (or should be) capable of doing.

## What does Blended Making look like?

Blended making can take on many forms. It has a loose definition that could be applied to many different project ideas. Rather than go into great detail on what is (or isn't) blended making, I will draw on three examples that I believe demonstrate the idea of blended making. That is to say, they all contain both physical and virtual elements that were designed to function together in some way.

## Example 1 - Flying Bicycle Game

**Fig. 1.1** Flying bicycle video game

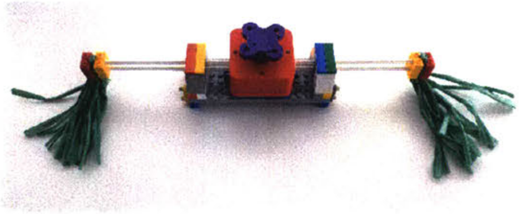


**Fig. 1.2** Flying bicycle handlebar controller

The first example is a flying bicycle video game. The virtual creation is a video game where the player moves a flying bicycle side-to-side to catch incoming tacos and balloons (Fig. 1.1). The physical creation is a set of handlebars constructed out of LEGO bricks and craft materials (Fig. 1.2). An accelerometer is incorporated into the design to allow the user to maneuver the bicycle on the screen by tilting the physical handlebars side-to-side. A button has also been built into the handlebars to trigger a bicycle horn sound effect in the game (because every flying bicycle needs a horn of course).

Blended making is demonstrated in this example through the design of a physical controller for interacting with a video game on the screen. The video game could have been controlled through keyboard input, but the process of creating a custom controller provides an opportunity to engage in design across multiple domains helping children to gain a deeper understanding of the design process. The designer must think about the characteristics that make up bicycle handlebars and attempt to construct a suitable replacement with available materials. This process provides another opportunity for the child to engage in iterative design beyond what they created in the virtual world.

## Example 2 - Wizard Story

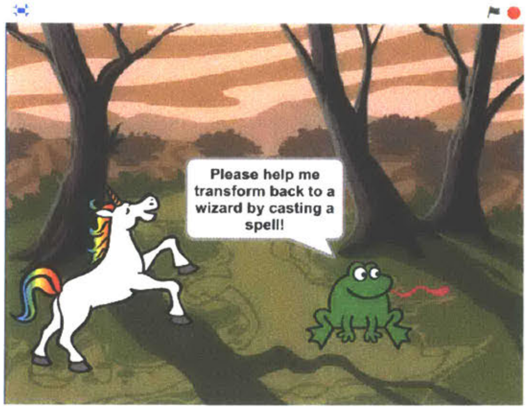


**Fig. 1.3** Wizard Story Scratch project

**Fig. 1.4** Wizard hat and magic wand

The second example is an interactive storytelling project. The story tells us that the main character's friend has been transformed into a frog and needs our help changing back into a wizard (Fig. 1.3). By wearing a custom wizard hat and magic wand (Fig. 1.4), the user is able to take on a role as a character in the story. Thrusting the wand forward causes a magic spell sound effect to play. Each time a spell is cast, the distressed friend changes to a different animal and makes a witty remark. When the correct spell is finally cast, things return to normal as the user and the virtual characters begin a celebratory dance party.

This example demonstrates blended making by combining virtual storytelling with physical costume design. Unlike the video game example, the physical actions (shaking the wand) simply progress through a series of predefined events in the story. Like turning the pages of a book, each shake of the wand progresses the story. The project could be designed to be more interactive by using different wand gestures to trigger various events in the story.

### Example 3 - Intruder Detector

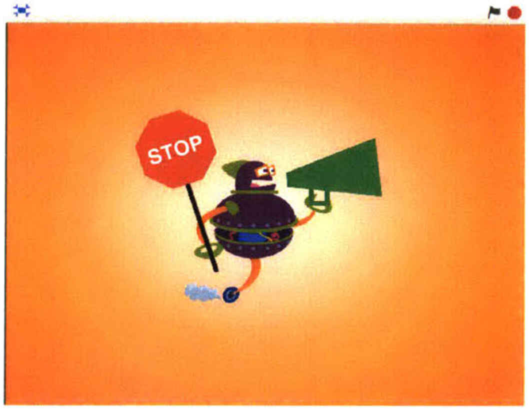


**Fig. 1.5** Intruder Alarm Scratch project

**Fig. 1.6** Motion activated door hanger

This final example demonstrates how we can use blended making to interact with the environment. Rather than creating a physical controller to be held or a costume to be worn, this project features a physical creation which has been embedded into the environment. The project uses a custom door hanger and motion sensor (Fig. 1.6). When the door is opened, an alarm is sounded and bright colors flash on the screen telling the intruder that they have been detected.

Even though the user isn't focused on the screen in this example, it still practices blended making. The virtual alarm system was created on the screen and connected to the motion-activated door hanger. Some children may prefer to place their focus on tinkering in the physical world rather than on a screen. One could imagine projects featuring music, games, or even pranks, where the virtual creation is in support of interaction in the physical world rather than the other way around.

## Technology for Blended Making

Today, there are a wide range of tools which can be used to engage in blended making. Often these tools are designed to promote either physical or virtual making, however by combining multiple tools or platforms one can create systems to engage in the type of making described in this thesis.

### Scratch

Scratch (Fig. 1.7) is a visual programming language and online community where anyone can create stories, games, music videos, tutorials, and a whole lot more. Scratch is developed in the Lifelong Kindergarten research group at the MIT Media Lab. The main goal of Scratch is "to support self-directed learning through tinkering and collaboration with peers" (Maloney et al., 2010). Through Scratch, young learners are able to learn skills like problem solving, critical thinking, and collaboration by sharing and exploring ideas that are particularly meaningful to them.

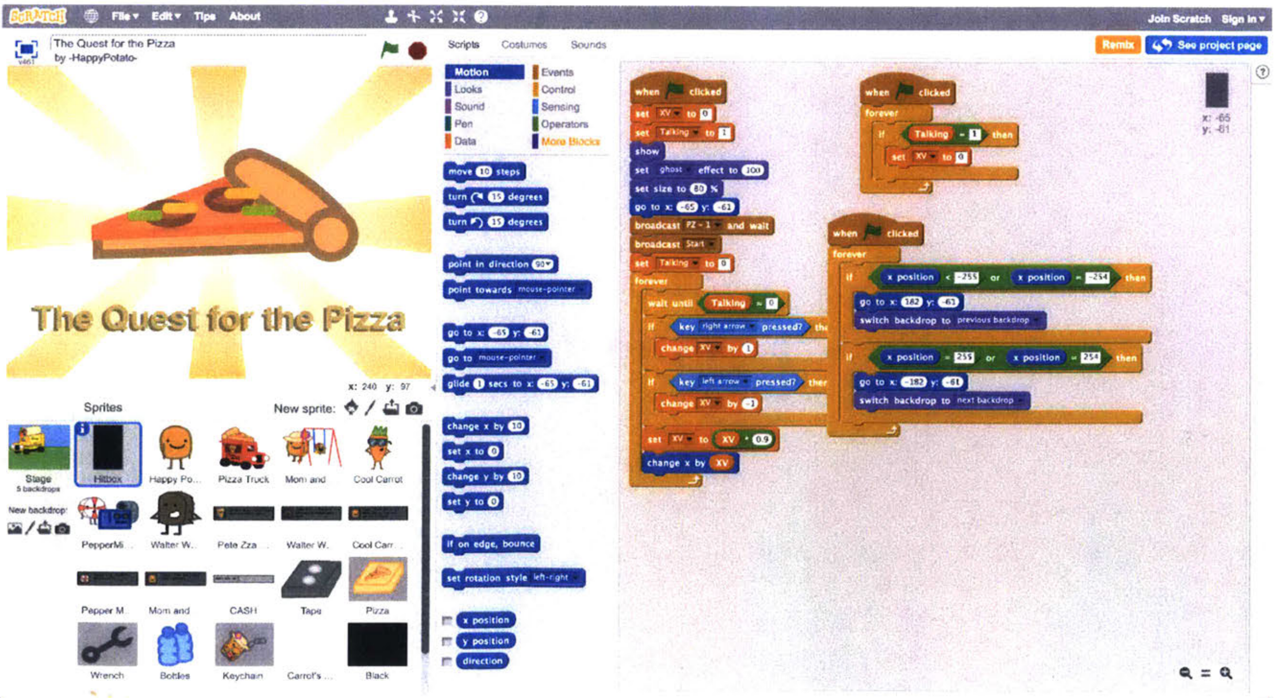


**Fig. 1.7** Scratch project titled "The Quest for the Pizza" by -HappyPotato-

Tinkering is a common way that children interact with Scratch. According to Mitch Resnick and Eric Rosenbaum, tinkering is characterized as “a playful, exploratory, iterative style of engaging with a problem or project. When people are tinkering, they are constantly trying out ideas, making adjustments and refinements, then experimenting with new possibilities, over and over and over” (Resnick and Rosenbaum, 2013). The Scratch project editor provides a wonderful opportunity for children to engage in the tinkering process by making programming scripts for different characters, playing with costumes and sounds, and testing and revising their ideas by pulling apart and snapping together different programming blocks.

## Scratch Extensions

Over the years, Scratch has seen a number of extensions to connect Scratch programming to the physical world. Some Scratch extensions have connected to new sensors or actuators for designing new physical interactions. Others bring in data from web services like live weather or International Space Station location information. Through many of these extensions, Scratch has long supported blended making.

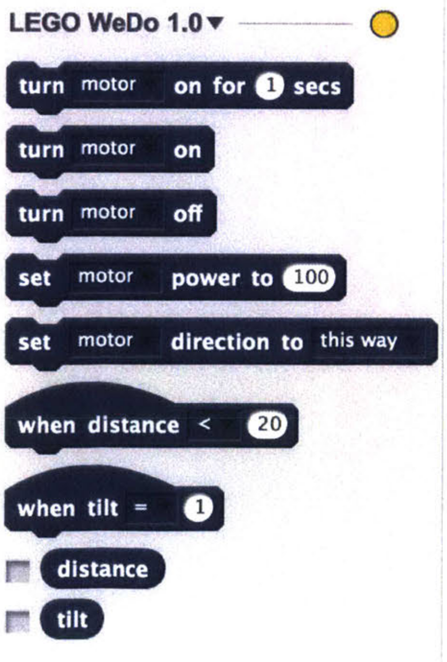
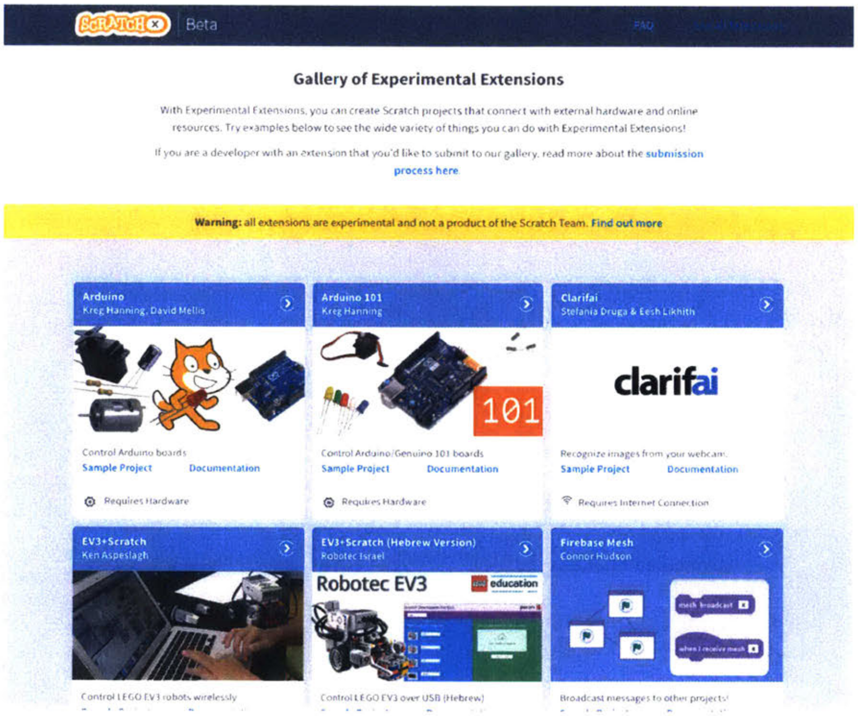


**Fig. 1.8** LEGO WeDo 1.0 Scratch blocks

A Scratch extension often takes the form of a new category of blocks that have a specific function. For example, the LEGO WeDo physical construction kit has long been supported by Scratch. The LEGO WeDo kit features light and tilt sensors and a motor for making physical creations that could connect with Scratch. By loading additional extension blocks (Fig. 1.8), Scratch gains the ability to incorporate sensor information or motor control into Scratch projects.



Gallery of Experimental Extensions

With Experimental Extensions, you can create Scratch projects that connect with external hardware and online resources. Try examples below to see the wide variety of things you can do with Experimental Extensions!

If you are a developer with an extension that you'd like to submit to our gallery, read more about the submission process here.

Warning: all extensions are experimental and not a product of the Scratch Team. Find out more

Control Arduino boards

Control Arduino/Genuino 101 boards

Recognize images from your webcam.

Control LEGO EV3 robots wirelessly

Control LEGO EV3 over USB (Hebrew)

Broadcast messages to other projects!

**Fig. 1.9** ScratchX extension gallery

In the past, many Scratch extensions were considered to be experimental additions to the Scratch block language. While there was a small set of extensions that were included in Scratch, other extensions were loaded through a hidden menu or in the case of Scratch 2.0, on a separate site called ScratchX (Fig. 1.9), designed with the specific purpose of letting people tinker with new and exciting extensions.

With the upcoming release of the next generation of Scratch, called Scratch 3.0, extensions will become more prominent. By exposing more extensions for tools and services that were designed with young learners in mind, I hope that this new system will allow a generation of new children to explore the ideas behind blended making.

## ScratchBit

Along with other members of the Lifelong Kindergarten research group, we have begun to implement a new dedicated hardware device for Scratch called ScratchBit (Fig. 1.10). The ScratchBit builds on the past of work of the Lifelong Kindergarten group including the last official Scratch hardware device, the PicoBoard (Millner, 2009).

**Fig. 1.10** ScratchBit design v3 with attachment modules

The ScratchBit is a wireless input device designed to work with Scratch. It provides a way of connecting Scratch to the physical world by exposing motion control through custom Scratch programming blocks. Through the use of an onboard inertial measurement unit (IMU), the ScratchBit can detect a series of high-level gestures (i.e. jump, shake, and move). In addition to the high-level gestures, the ScratchBit also exposes some lower-level information such as a two axis tilt angle or ambient light level.

The ScratchBit is designed to be easily attached to a wide range of objects such as dolls, skateboard, pants pockets, and swing sets. By designing the ScratchBit to seamlessly attach to a wide range of materials, it will hopefully support tinkering no matter what the user is passionate about. The wireless connectivity adds to the ScratchBit's ability to be seamlessly integrated as a physical construction kit for Scratch. It can be used at a range of approximately 60 feet (20 meters) depending on the environment. This makes the ScratchBit a useful tool regardless of whether it's being used in a bedroom, classroom, or playground.

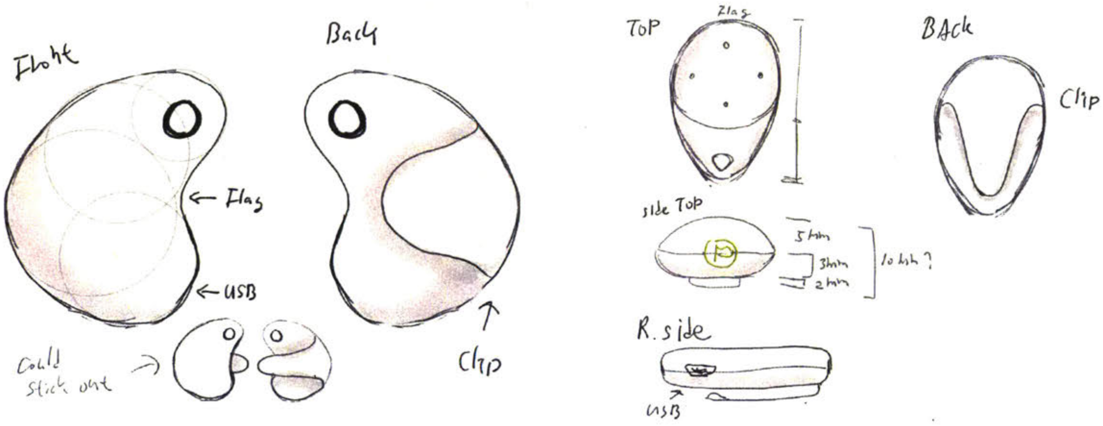


**Fig. 1.11** Early ScratchBit mockups

# 2. Inspiration and Related Work

Inspiration for the ScratchBit has come from various construction kits, both physical and virtual. This chapter will describe several different platforms that have provided a source of inspiration for the development of the ScratchBit and the ideas behind blended making. A description of each platform will be provided as well as how it influenced the ScratchBit project.

## PicoCricket

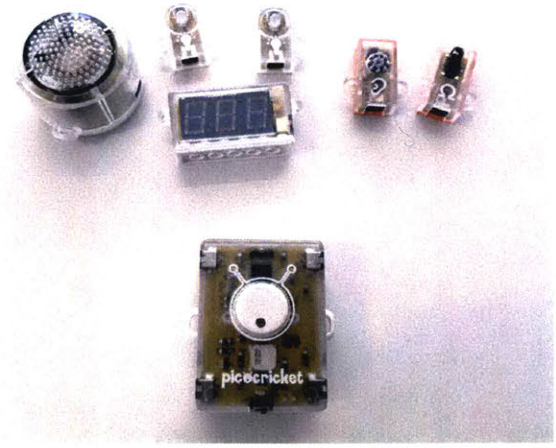


**Fig. 2.1** PicoCricket device with input/output peripherals

The PicoCricket (Fig. 2.1) was a robotics construction kit that attempted to merge art and technology. The PicoCricket attempted to do this by "enabling young people to create artistic creations involving not only motion, but also light, sound, and music" (Rusk et al., 2008). The

PicoCricket technology was originally developed by the Lifelong Kindergarten group and later commercialized by the Playful Invention Company.

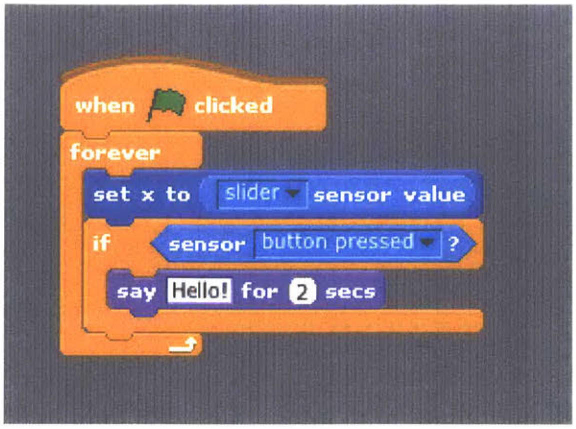
The PicoCricket used a series of outputs: multi-colored light, sound box, numerical display, and motor, as well as a series of inputs: light sensor, touch sensor, sound sensor, and resistance sensor to provide children with unique ways of interacting with the physical world. Similar to the ScratchBit project, the PicoBoards sensors and actuators were pre-selected with the intent to support a wide range of project ideas.

To use the PicoCricket, children would use a block-based programming language, similar to Scratch, which allowed them to snap together programming blocks to control how the PicoCricket would operate in the physical world. An important distinction between the PicoCricket and some of the other tools I will discuss is that the PicoCricket program did not connect back with the physical screen. That is to say, once you have programmed the PicoCricket, it would operate only in the physical world. Unlike Scratch, the PicoCricket software did not feature a graphics rendering engine.

One clear similarity between the PicoCricket and the ScratchBit was the emphasis on constructing with craft materials. In addition to various LEGO components, the PicoCricket came with craft materials such as: pipe cleaners, string, foam balls, and pom-poms. By encouraging construction using craft materials in addition to the electrical and mechanical components, the PicoCricket could be used as a general purpose physical construction kit for things like storytelling and interactive art projects.

## PicoBoard

The PicoBoard (also known as ScratchBoard) can be thought of as the spiritual predecessor to the ScratchBit. The PicoBoard was developed by the Lifelong Kindergarten group over a decade ago. Similar to ScratchBit, the PicoBoard featured a set of sensor inputs designed to connect with a custom set of Scratch programming blocks (Fig. 2.2). The PicoBoard interfaced with Scratch through a physical serial port connection.



```
when [flag] clicked
forever
  set x to (slider sensor value)
  if <sensor button pressed ?>
    say Hello! for 2 secs
```

**Fig. 2.2** Scratch 1.4 programming blocks for PicoBoard

The physical design of the PicoBoard featured a large flat circuit board (Fig. 2.3) with exposed light and sound sensors, a momentary push button, and a slider. It additionally featured four connectors for creating simple switches with conductive materials. The circuit board was silkscreened with playful iconography such as an ear near the sound sensor and the light sensor appearing as the pupil of an eye. These designs decisions seem to add an element of discovery to the physical design. A child would most likely identify the playful imagery before discovering the adjacent sensor.

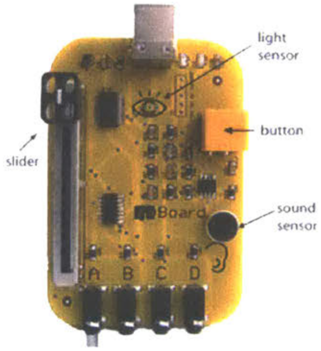


**Fig. 2.3** PicoBoard c. 2007

The ScratchBit is in many ways related to the PicoBoard. They are both input only devices, used for manipulating assets (e.g. sprites, variables, sound effects) within the Scratch environment. Unlike the PicoBoard, the ScratchBit is able to control Scratch wirelessly via bluetooth communication. Since the ScratchBit places an emphasis on embeddability, it is fully encapsulated in a rigid enclosure with explicit attachment mechanisms. This is in contrast to the bare circuit board design of the PicoBoard.

## Makey Makey

In my opinion, Makey Makey has been one of the most successful devices to enable blended making. Makey Makey lets you turn everyday objects, like bananas and aluminum foil, into custom interfaces for controlling computer software. The video on the Makey Makey website (Makey Makey, 2018) demonstrates an array of inspiring projects from a piano staircase to a cat selfie photo booth (Fig. 2.4).

**Fig. 2.4** Cat selfie booth made with Makey Makey

Similar to the PicoBoard, the physical design of the Makey Makey is an exposed circuit board. To use the Makey Makey, the user must connect conductive materials to a series of exposed pads via a set of included alligator clips (Fig. 2.5). The Makey Makey uses closed electrical circuits to trigger keyboard events on the computer (Silver and Rosenbaum, 2012). This provides a very accessible platform for constructing interfaces with objects in the physical world.

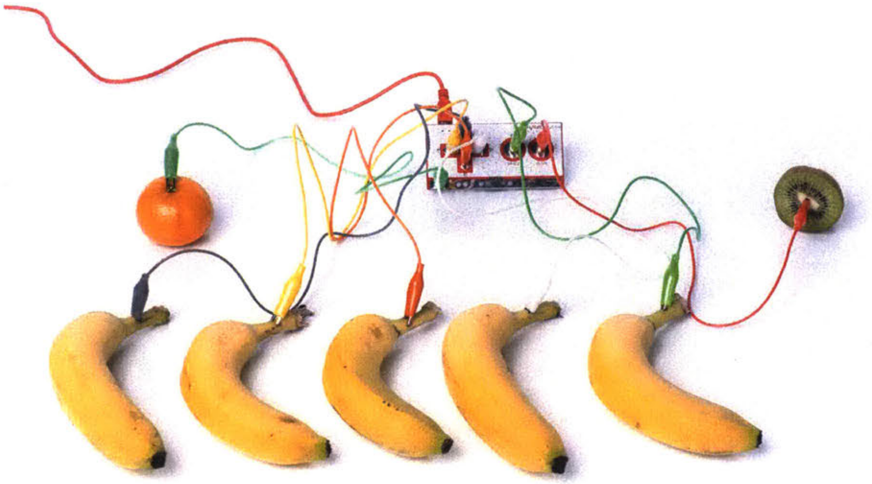


**Fig. 2.5** Makey Makey fruit piano interface

The Makey Makey was not designed to work with any particular software. Instead, when it's connected to the computer's USB port it is recognized just like any other USB keyboard. Therefore almost any software that can be controlled with a keyboard can be controlled with the Makey Makey. By itself Makey Makey is a wonderful physical construction kit. But, it does not require that the user constructs the virtual interface. When paired with a virtual construction kit like Scratch, the Makey Makey becomes an indispensable tool for engaging in blended making.

The ScratchBit takes a cue from Makey Makey's ease of use. With the Makey Makey, it's easy to get started without much prior knowledge of the system. By simply tapping your fingers on the Makey Makey, you can start triggering certain actions in Scratch. As you begin to understand the role conductive materials play in the physical construction, you can begin to construct more advanced interfaces. Similarly, the ScratchBit employs a set of high-level gesture blocks (i.e. jump, shake, and move) to provide an easy way to get started. The user can attach the ScratchBit to their pocket and trigger actions by moving around. As they gain some familiarity with the ScratchBit, they can create more elaborate interfaces by attaching the ScratchBit to other objects or by building into their own creations.

# 3. Design Principles

I created a set of design principles to evaluate the success of each ScratchBit design iteration and also to help guide the development as we began preparing to bring the device to market. The four design principles are:

1. Easy to Get Started
2. Support Blended Making
3. Emphasize Embeddability
4. Keep the Cost Low

## Easy to Get Started

One of the reasons Scratch has been successful in reaching so many children around the world is the emphasis on the beginner experience. The Scratch community provides many opportunities for new users to start engaging with others for support and inspiration. Getting started resources such as the Scratch Coding Cards (Fig. 3.1) and online tutorials, created by Natalie Rusk and others on the Scratch team, help Scratch users discover new features while creating projects that span a wide range of different interest areas.

**Fig. 3.1** Scratch Coding Cards

The Lifelong Kindergarten research group often refers to this as focusing on low floors. If you spend enough time in our research lab, you will hear a reference to low floors, wide walls, and high ceilings. This design philosophy allows us to think deeply about the beginner experience (low floors), providing many different ways of using the tool for personal expression (wide walls), but also ways for them to work on increasingly sophisticated projects over time (high ceilings) (Resnick, 2016). Lately, the Lifelong Kindergarten group has placed more emphasis on the low floors and wide walls. I personally welcome this as I see an increasing number of platforms that provide (and often front load) loads of complexity often to the detriment of autonomy.

The ScratchBit project carries on this same emphasis on low floors and wide walls. The high level gesture blocks and sensor abstractions provide an easy way to start using the ScratchBit. A diversity of example projects like game controllers, musical instruments, and art installations attempt to inspire a wide range of different interests.

While we have emphasized the low floors and wide walls, we have not forgotten about the high ceilings. Providing access to lower level sensor information such as tilt angles and analog ambient light readings allow for children to create more complex projects as they become more comfortable with the ScratchBit. In addition to technological complexity, integrating the ScratchBit in physical constructions provides ample opportunity to go beyond the beginner experience.

In a later chapter I will also discuss the getting started resources that were designed during the development of the ScratchBit. Like the Scratch Coding Cards, these resources proved to be useful in introducing the device to children for the first time.

## Support Blended Making

The ScratchBit's purpose is to present a physical construction kit that provides a seamless integration with the Scratch programming environment. By itself, the ScratchBit serves no function. It cannot be programmed to operate autonomously from a computer. However, by pairing the ScratchBit with Scratch, we hope to provide an easy entry point into blended making for creators of any age.

One of the features that makes Scratch so tinkerable is the ability to move from coding to testing and debugging in one seamless process. With many existing physical construction kits, the physical creation is somehow sent code from a computer and then operates autonomously. Instead, without having to download code each time you run a project, the ScratchBit allows the user to seamlessly move back and forth between physical and virtual creation.

## Emphasize Embeddability

An emphasis on embeddability (or attachability) is one of the key features that differentiates the ScratchBit from other platforms. In a later chapter, I will discuss how we have designed three unique attachment modules to provide easy ways to attach the ScratchBit to a wide range of materials. In addition to making attaching the ScratchBit to materials easier, the attachment modules provide explicit affordances that will hopefully inspire diverse creations in the physical world. In our design meetings, we would often brainstorm about what kind of materials children might use to construct with. In the end we decided that to best support creativity, the ScratchBit needed to work with materials ranging from recycled cardboard, to sneakers and hats, and even toys and sports equipment.

My colleague Jie Qi and I often joked that the ScratchBit design needs to be complete, but feel incomplete. What we meant by that was that the device should feel like a finished object while holding it in your hand. This meant the final design was not going to be an exposed circuit board. Although we wanted it to feel like a finished object in the hand, we still wanted it to feel a little boring that way (i.e. incomplete). We hope that the design would inspire the user to stick it to existing objects or build new, unique creations with it.

## Keep the Cost Low

While this design principle wasn't particularly useful in evaluating ScratchBit workshop structures, it was a constant guiding principle throughout the development of the ScratchBit prototypes. Scratch is a global community. It is currently being used in nearly every country around the world. Like Scratch, we also want the ScratchBit to be able to reach the global community.

Instead of investigating the latest cutting-edge sensors and technology, we have intentionally selected electronics that are widely available. With the increasing number of smartphone devices in production, the cost of inertial measurement units (IMU) used for measuring motion has dropped considerably. In addition to the wide flexibility they offer in terms of human interaction, the mass production of IMUs made them an ideal candidate for the ScratchBit.

Unlike virtual programming blocks, hardware like the ScratchBit costs money to reproduce. From manufacturing to distribution each additional ScratchBit means raw material and labor costs. To me, this is one of the reasons virtual creation platforms like Scratch are so equitable. Scratch is free whether you use ten or a thousand programming blocks. Unfortunately, we don't have unlimited resources to produce and distribute ScratchBits at no cost. We are, however, trying to keep manufacturing costs as low as possible while not compromising on the other design principles.

# 4. Play Testing

Workshops, or play tests as I will refer to them, were conducted frequently and with a diverse population. I like the term "play test" as I think invokes a more fun and playful spirit than workshop. Testing with young children, teenagers, university students, educators, and tinkerers of all ages was essential for evaluating both the effectiveness of the ScratchBit and the overall structure of the play test.

As we conducted play tests, the need to develop certain resources and materials was essential for making sure it was easy to get started using the ScratchBit. In this chapter, I will discuss some of the getting-started resources and materials that were developed as a result of the various play tests.

In Lifelong Kindergarten, we will often come up with a unique theme to frame each play test. This often helps to provide an clear entry point rather than starting with a blank canvas and seemingly limitless possibilities. But, with the ScratchBit, I wanted to observe how participants explored their own interests without restricting types of projects that might be created.

I needed to find another way to provide this entry point without biasing (too much) how the device might be used. In her book, "The Having of Wonderful Ideas: and Other Essays on

Teaching and Learning", Eleanor Duckworth speaks about where wonderful ideas come from: "Wonderful ideas do not spring out of nothing. They build on a foundation of other ideas." (Duckworth, 2006). To attempt to build a foundation for others to explore their ideas in the physical world with the ScratchBit, I collected (and continue to collect) a set of materials that I will refer to as *wonderful objects*. These wonderful objects were useful in a workshop setting, but you could imagine children using the ScratchBit to explore their own objects of personal significance. Or maybe it will allow them to "fall in love" with new objects as Seymour Papert described in his 1980 book, Mindstorms, Children, Computers, and Powerful Ideas (Papert, 1980).

The workshop structure would usually go as follows:

- Brief introduction
- 5-minute ice breaker
- Demonstration of ScratchBit example projects and "live" coding
- 10-minute warm up activity
- Quick share out
- 45 - 60 minute free play
- Final share out and reflection

## Getting Started Resources

One of the most important aspects of the workshop structure was the "getting started" experience for the participants. This would include showing example projects that could inspire the participants, but also making sure to show a diversity of projects to not bias them into only thinking about the device in a certain way (e.g., a game controller).

I came up with three example projects, outlined earlier in the "What does Blended Making Look like?" section of Chapter 1, that I hoped would demonstrate a diversity of projects: a game controller, a story telling costume, and a bedroom alarm system. I would often show these examples while introducing the device to a new audience leaving time to "live" code a simple example together.

Together with a few other members of the Scratch team, including designer Carl Bowman, we came up with a set of ScratchBit mini-cards to demonstrate several simple ways you can use the ScratchBit to build a custom interaction with Scratch. The ScratchBit mini-cards (Fig. 4.1) mainly focused on the high-level gesture blocks, but also included some more advanced techniques which made use of the analog input blocks (called reporters in Scratch).

**Fig. 4.1** ScratchBit mini-card examples

## Wonderful Objects

As I mentioned earlier, I slowly amassed a large collection of objects to inspire play test participants to use the ScratchBit in many different ways. I will refer to these items as "wonderful objects" as their purpose is to provide a foundation for which participants can explore their own ideas.

Most of the objects I obtained came from a local dollar store. I have earned a reputation around the research group for my dollar store shopping prowess. I can't help but get inspired as I walk through aisles finding toys, household objects, and random seasonal trinkets for constructing in the physical world. I find myself contemplating new blended making projects each time I venture to my local Dollar Tree as it's called.

**Fig. 4.2** A collection of wonderful objects from a recent dollar store venture

From plastic flamingos, to kitchen utensils, to sports equipment, and even pet toys, these wonderful objects are selected to challenge the participants to think about how the ScratchBit could be used to construct things that are personally meaningful to them.

# 5. Design Iterations

In the Lifelong Kindergarten research group we are inspired by how children learn in kindergarten. In Kindergarten, children learn by exploring ideas in a highly iterative process. They explore colors by mixing paint with their fingers, storytelling through pretend play and dressing up, and about structures as they build with clay, blocks, and craft materials. In doing this they engage in what Mitchel Resnick calls the Creative Learning Spiral (Fig. 5.1).

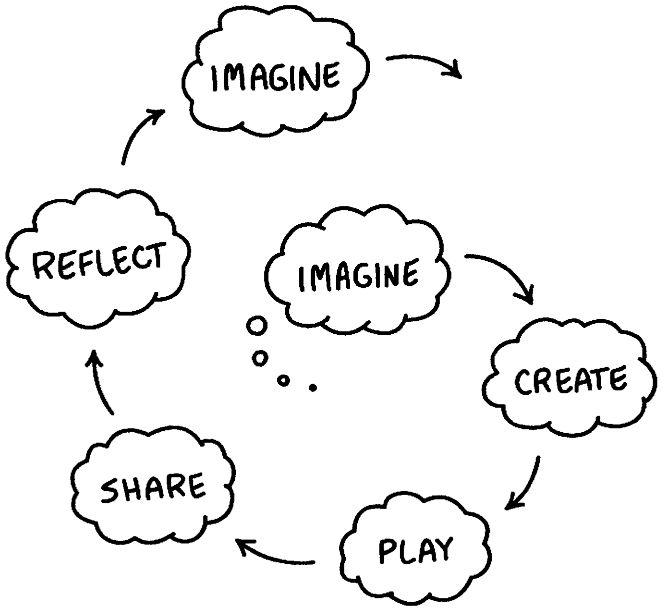


**Fig. 5.1** The Creative Learning Spiral

As children in Kindergarten engage in the creative process they often "imagine what they want to do, create a project based on their ideas, play with their creations, share their ideas and creations with others, reflect on their experiences – all of which leads them to imagine new ideas and new projects." (Resnick, 2007).

The way our team engages in the iterative design of the ScratchBit is no different. As we move between each design iteration, we practice each step of the Creative Learning Spiral. We *imagine* by coming up with a initial concept or a set of design principles that will begin to guide the project. Next we start to implement the idea either by adapting existing technologies or developing our own prototypes. A lot can be learned during the *create* stage. We often discover physical or technical limitations, stumble upon new ideas, and begin to think about how we might introduce this new technology to children and educators. Then we arrive at what I personally find to be the most fun stage of the creative learning spiral: *play*! We usually start with small internal workshops with members of the Scratch Team and Media Lab community. Towards the end of the workshops we always save time to *share* out what we have been working on. This provides an opportunity to *reflect* about overall workshop structure. Reflecting on the workshop provides useful insight into how we introduce the technology, the types of example projects we demonstrate, and what kind of resources or prompts can help to get people started.

Reflection is such an important part of the creative learning spiral that it deserves a paragraph of its own. In addition to reflecting at the end of each major design iteration, countless smaller reflections are done all throughout the process. We might conduct many workshops over several weeks or months with the same design. At the end of each workshop, the facilitators meet for a debrief and discussion. We usually organize our reflections in a red, yellow, green structure. Red indicates some major issue that needs to be addressed, yellow means it needs some work or raises some questions, and green is for things that went really well. The discussions are by no means limited to technology issues; we often spend more time reflecting on things like participants' language, the types of projects that were created, how easy it was for people to get started, etc. Each week the group of people working on the ScratchBit meets to discuss the results from the play test and think about future directions. Through this sort of continuous reflection, we hope to continue to guide the project forward while staying true to our design principles.

## Hardware Design Iteration 1

After several weeks of discussions around what sensors our new device might include and how it might be used for controlling Scratch physically, the maker in me was itching to implement at least a basic prototype. I have always been able to think best when I have a tangible object to tinker with and manipulate. Using the many tools of the MIT Media Lab, I set out to create a prototype that would allow me to think about this new device in a tangible way.

**Fig. 5.2** ScratchBit design iteration v1

The first design iteration (Fig. 5.2) was built around a popular bluetooth development board made by Adafruit. If an existing technology can easily be adapted to meet the needs of the prototype then it makes sense to use it. Only later did the ScratchBit's design necessitate the use of a custom circuit board. For sensor inputs, the prototype featured an accelerometer for providing tilt detections and a potentiometer mounted on top of a single momentary switch providing both analog (potentiometer) and digital (switch) inputs. The 3D printed enclosure had four LED indicators for showing the user the current tilt direction. A pair of "ears" on either side of the case were added to assist with strapping the prototype to various materials. The top of the design featured a large rotation knob with clicking action. The rotation knob was outfitted with loops and hooks for using craft materials like string or pipe cleaners to attach other larger objects (imagine a cardboard steering wheel attached to the rotation knob for a driving game). The loops were also spaced apart at a multiple of 8mm making it compatible with the LEGO's Technic system.

This prototype served its purpose well. After running several play tests, patterns began to emerge around how people used the device for physical construction which was valuable in evaluating the prototype based on the set of design principles. The prototype was succeeding in supporting blended making, we were seeing new projects made with craft materials and wonderful objects. One project used a bicycle wheel as an interface for a spinning color wheel in Scratch by attaching the prototype to an actual bicycle spoke (Fig. 5.3).

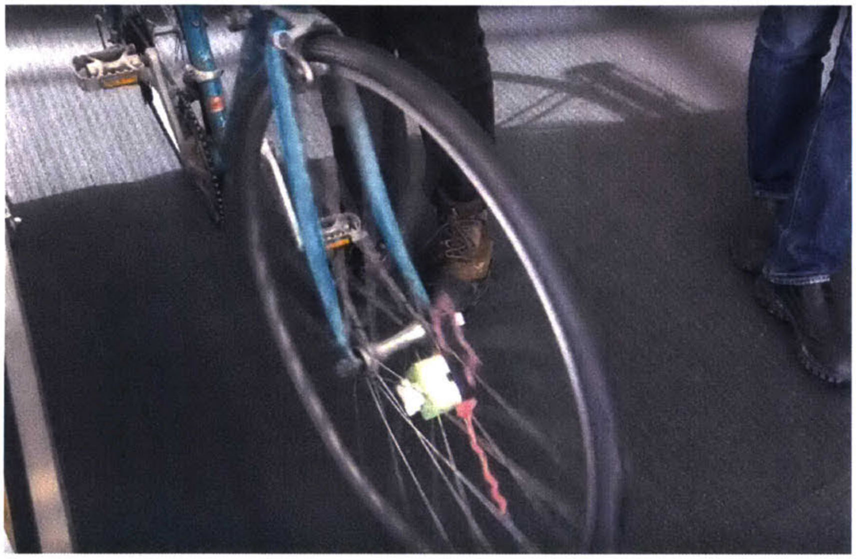


**Fig. 5.3** ScratchBit prototype v1 attached to bicycle spoke

There were, however, still some challenges in how easy it was to get started and the overall embeddability. The large size of the device didn't work so well with the ears being the only attachment point. It was often difficult to strap the prototype securely to the intended object or material. We also gained valuable insight into how people thought about the prototype by observing the types of projects being created and which of the programming blocks were being used. This information was used not only to iterate on the block language, but also to start to think about what kind of getting started resources and example projects would be useful to inspire a diverse use of the ScratchBit.

While I enjoy doing basic design work, my expertise is not in industrial design. Fortunately, we decided to contract industrial designer Dan Chen to help with the next round of prototype. Not to mention Andrew Sliwinski, co-director of the Scratch project, was also helping to move us to the next prototype with his keen design training.

## Hardware Design Iteration 2

As we moved into the second design iteration, it became clear that prebuilt prototype platforms would no longer be able to support our industrial design. After some discussion around the different platforms we could use for our next prototype, we landed on the popular Nordic NRF51822 line of bluetooth SoC's. While we had a specific idea in mind about how this next prototype would function, the design principle of *Keep the Cost Low* was always part of the discussion.

Having attachment points on both the rotation knob and the base of the original prototype proved to be confusing. The ears on the base were used for attaching the ScratchBit to objects (bicycle spoke) and the knob was often used for attaching objects to the ScratchBit (paper plate steering wheel). Some play test participants even voiced their concern of how it wasn't clear which part they should be using for attachment. To attempt to address this issue, we began sketching out various ways to incorporate clipping mechanisms into the design.

**Fig. 5.4** ScratchBit prototype v2

After weeks of back-and-forth discussions, we finally landed on our second ScratchBit design iteration. This prototype would feature a large clip at the bottom of the base for attachment. The clip could be used to quickly slot the ScratchBit to materials like pants pockets, cardboard, baseball cap visors, etc. We optimized the slot to grip cardboard, a widely available and highly composable material. In addition to the clip, we slimmed down the rotation knob on top, but preserved its use as a knob and a button. The tilt indicators in this prototype would cast light through the top of the rotation knob while activated. This version also included a new feature: a small "green flag" button on the side of the case. In Scratch, the green flag is often used to start (or restart) a project. By including this button, the ScratchBit would allow users to start their projects from a distance.

## Hardware Design Iteration v2.1

While this prototype helped simplify the embeddability considerably, another issue arose. Attempting to use the rotation knob and the tilt sensors at the same time proved to be an awkward maneuver. The only way to use both sensors at the same time was to hold the device in your hand. While we certainly expected that some children would use the ScratchBit like a game controller, it wasn't the style of play we were designing for. The design principle of *Emphasize Embeddability* stated that the ScratchBit should encourage the user to attach it to existing objects or integrate it into new constructions. Unfortunately, embedding the ScratchBit often resulted in losing easy access to the rotation knob. We saw this as a problem that needed to be addressed. This forced us to re-examine whether or not a rotation knob was a type of sensor that could support our design principles.

By shifting a few things around, we were able to create a mini design iteration, v2.1 (Fig. 5.5). We replaced the rotation knob with stationary lid and this time included a sticker of the Scratch Cat to help with the directional orientation. Unfortunately, dropping the rotation knob also meant dropping the push button. There was hesitation from some members of our team, but we decided this intermediate step was necessary even if we decided to bring the button in a future hardware revision.

**Fig. 5.5** ScratchBit prototype v2.1

By dropping the rotation knob we were losing what some thought was a valuable analog input sensor. To address this, we drilled a small hole into the side of the case to include an analog light sensor. Light sensors have long been a favorite of our research group. They add an element of surprise by being able to make something happen on the screen by waving your hand in front of the device.

## Hardware Design Iteration 3

At the time of writing this thesis we are now on our third, and what might be final, major design iteration. After we decided to drop the rotation knob, there was no longer a need for the device to be as thick as it was. Rotation knobs that have a built-in momentary switch are prohibitively expensive, especially if we were to achieve our fourth design principle of *Keep the Cost Low*. This forced us to design our own system to achieve a clickable knob. Once we were no longer designing for this, we were were able to reduce the height of the device considerably (Fig. 5.6).

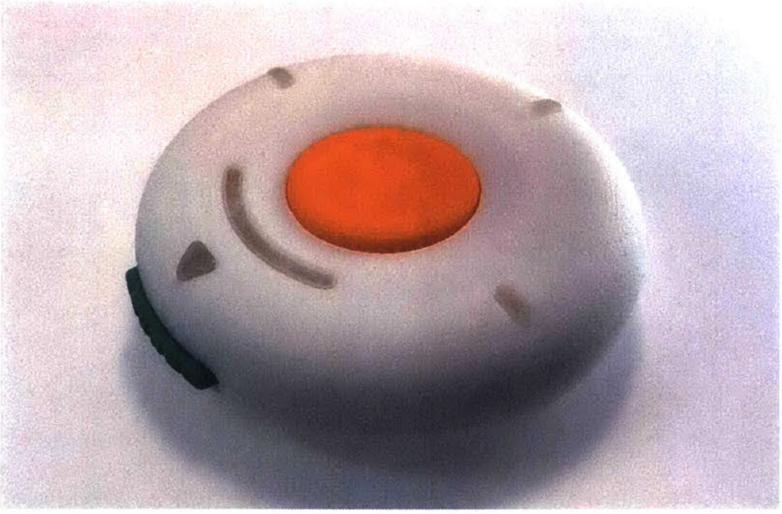


**Fig. 5.6** ScratchBit prototype v3

The need for a button that could be used in the project rather than just at the start of the project (green flag button) forced us to bring the top button back. This time, we placed it prominently in the top-center of the ScratchBit. The light sensor was integrated into the case design mounted directly beneath the arc surrounding the button. The green flag button also received a redesign, making it more prominent on the side of the case.

Beyond the new slim design, one other major difference in this version of the prototype was creating three discrete attachment modules for wrapping around the base unit (Fig. 5.7). Each attachment module serves a specific function. After quite a bit of discussion on the correct types of attachments we landed on the following three. *Clip it*, the large clip that we had found so useful in v2 of the prototype would be one. *Tie It*, would feature four holes for easy looping with string and pipe cleaners. An included elastic cord with cord clamp, provides an clear example of how this attachment is to be used. And finally *Stick It*, would be the last attachment which featured a single large suction cup for quickly sticking the ScratchBit to smooth, flat, surfaces.

**Fig. 5.7** ScratchBit prototype v3 with attachment modules

The base unit could still be used in hand or without an attachment, but we hoped that the three different attachment modules would add a clarity in how you might start thinking about different ways to attach the ScratchBit to materials.

This design iteration has proved to be very successful in meeting most, if not all, of the design principles. Unfortunately, the different attachment modules will increase the manufacturing costs, but we are hopeful that their usefulness will far outweigh the cost.

## Software Iterations

Another key part of the development of the ScratchBit was the programming blocks. This is an area that has seen quite a bit of iteration throughout the design process. Equally as important as the physical affordances, the programming blocks for using the ScratchBit needed to make it clear how the ScratchBit could be integrated into Scratch projects while still allowing them to be used in a wide range of different projects. An early version of the block palette (Fig. 5.8) featured various programming blocks for interacting with the button, rotation knob, and the tilt sensor.

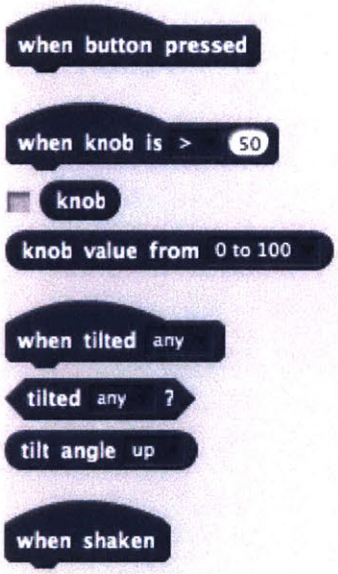


**Fig. 5.8** Early block palette for ScratchBit

The button was exposed with a single “when button pressed” event block (known in Scratch as a “hat block”) that was used to trigger events in Scratch each time the button was pressed. Some simple examples of how this this block was used were attaching a sound effect (e.g. cat meow) so that each time the button was pressed a sound would emit from the virtual character or using a change graphic effect block to change how a character looks on the screen each time the button is pressed.

For the rotation knob, we created another hat block for trigger events. Unlike the button hat block, this one had a settable threshold so the block could be triggered each time the knob was turned above or below a certain value. Keeping the beginner experience in mind, we sometimes think about using predefined thresholds rather than allow the user to set their own. In the case of the rotation knob, we decided at the time that a settable threshold made the most sense. We also created a block that could return the analog value of the rotation knob labeled, “knob”. By default, this block would return a value between 0 and 100 depending on the current state of the rotation knob. In addition, we included a block, “knob value from [0 to 100]”, that allowed you to choose the range of the analog rotation knob value. Four predefined ranges were included: 0 to 100, 0 to 360, -240 to 240, and -180 to 180. These ranges were selected to complement existing features in Scratch. For example, 0 to 360 worked well for rotating characters on the screen, and -240 to 240 (x-axis) and -180 to 180 (y-axis) matched the cartesian coordinate system that the Scratch graphical renderer employs.

The tilt sensor was exposed through a few different styles of blocks. A hat block was used to trigger events when the ScratchBit was tilted in a certain direction. One major issue this presented was that the ScratchBit didn’t have a strong directional orientation. In this early software iteration, the tilt directions were referred to as front, back, left, and right. In the hand, the ScratchBit lacked a clear front and back, and this became even more of an issue once it was embedded into other objects. This is an issue that we attempted to address by designing a new custom block menu which I will discuss later. Similar to the hat block, we included a boolean block, “tilted [any]?”, which could be incorporated into if/else statements in Scratch projects. Finally, we included another analog block (known as reporters in Scratch) that would provide an accurate tilt measurement across the x and y axis of the ScratchBit. This analog block was used extensively for making projects that would fluidly move a character or objects around the screen in one or two axis.

The strategy for the tilt sensor was similar to what had been done with the LEGO WeDo construction kit. Like the ScratchBit, it too included a tilt sensor which could be used to detect tilting across two axis. While we were happy with this strategy, we wanted to make sure it was easy for children who were new to the ScratchBit to get started using the tilt sensor. After some brainstorming sessions we came up with a the idea of including a "when shaken" hat block that would allow the user to simply shake the ScratchBit to trigger interactions in Scratch. This new gesture block made it easy to make projects like magic wands that could be shaken to cast spells, or as with one example, a bug squashing game where the ScratchBit could be affixed to the players shoe so they could stomp to squish virtual bugs in a Scratch project. After play testing this new gesture block, we discovered that it provided a much easier way to get started with motion control, leaving the more advanced tilt blocks to be discovered later.

## Gesture Blocks

With the success of the "when shaken" block, we started to brainstorm other possible gestures that the ScratchBit should support. Like the "when shaken" block, we wanted the other gestures to remain fairly high-level. Instead of attempting to detect very specific patterns (i.e. imagine drawing certain shapes in the air with the ScratchBit), we set out to create gestures that could be used many different ways.

**Fig. 5.9** ScratchBit Gesture Blocks

After another brainstorming session around possible gesture ideas, we came up with what we felt were three general purpose gestures: moved, shaken, and jumped (Fig. 5.9).

The "moved" gesture detects any slight movement. If you knock on the surface next to the ScratchBit or attach it to an object that moves even in the slightest, the "moved" gesture will continuously trigger. This highly sensitive gesture is useful to tell if an object is in motion or if something has been shifted around (think bedroom alarm system).

"shaken" carried over from the earlier software iteration. It proved to be very useful in a wide range of projects both attached to the body and to physical objects. Like "moved", shaken tracks general movement, but it requires more acceleration. The user must move the ScratchBit much more aggressively to trigger "shaken" compared with "moved".

The last gesture is "jumped". For those who have worked with motion sensors before, "jumped" is best described as freefall detection. It triggers each time the ScratchBit is tossed into the air, or, if you attach it to your pocket, by physically jumping around the space. We've seen jumped be used for a range of projects from body controllers where you have to jump to control a character on the screen to a hot potato game where the ScratchBit was embedded inside of a physical ball to detect if it was (or wasn't) tossed in time.

We will continue to think about implementing new general purpose gestures in the future, perhaps as the arise through play testing. But for now this set of three gestures has proven to be very useful to inspire many different uses of the ScratchBit.

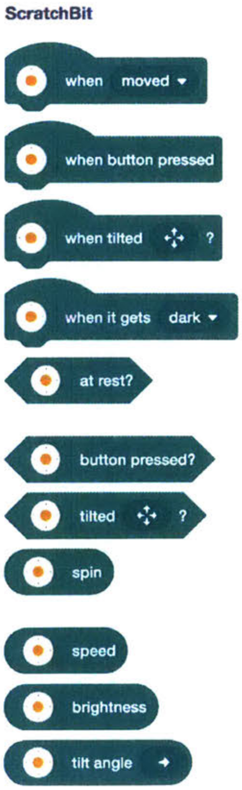


**Fig. 5.10** ScratchBit block palette as of writing this thesis

# 6. ScratchBit Projects

This chapter will describe three different projects that were created by children during ScratchBit play testing. Each of the children had very different backgrounds and interests as is evident in the projects they created.

## Suzanna, Age 11

Suzanna's explorations with the ScratchBit were one of the most interesting case studies for me. From the moment I first met her, one thing was very clear: Suzanna liked sports. I recall on our first session Suzanna was wearing a jersey from her favorite athlete, Tom Brady, quarterback of the New England Patriots.

Suzanna took part in a six-week recurring ScratchBit play test that I offered as an afterschool class at a local elementary school. Each week, I would travel with one of my colleagues, Jaleesa Trapp, to the school with batch of ScratchBits and wonderful objects. We wanted to observe how the ScratchBits were used over time, once they became a more familiar material. Most of the past play tests had been conducted in single one-hour to two-hour session. While observing the beginner experience was important, it was also important to see what sort of patterns, or fluencies, started to emerge over time.

When I first asked the question, "Has anyone used Scratch before?", each student took a turn talking briefly about their experiences with Scratch. When it came to Suzanna, she shrugged off the question with, "Yeah I've used Scratch before". I asked what kind of project she had made using Scratch, to which she explained that her class all went to the library to learn Scratch and complete a maze activity together. To Suzanna, she had “done Scratch”. She thought of it as an item to check off a list instead of as an open platform she could use to express herself and her ideas.

It wasn’t that Suzanna didn’t like Scratch. When she did the maze assignment at the time it was something new and perhaps exciting for her. She just didn’t see much use for it beyond that initial experience.

When we got started using the ScratchBits, Suzanna went right to the wonderful objects and quickly discovered a foam ball. She started using tape to attach the ScratchBit to the ball, but quickly became discouraged as the tape wouldn’t adhere well and the ScratchBit was throwing off the ball’s equilibrium. I dropped a little hint by saying, “It would be nice if we could get the ScratchBit inside the ball.” After a moment of deliberation, Suzanna looked up with wide-eyed excitement and asked, “Can we cut into the ball?”.

Once the ScratchBit was firmly inserted inside of the ball, Suzanna was off. She started exploring the programming blocks to see what kind of information she could capture about the ball. She started with a hot potato game where you would have to throw the ball back and forth with a partner before the timer went off. By the end of the first session Suzanna was brimming with ideas. On her way out of the classroom, she even asked if she could bring things from her bedroom so she could use them in her game during the next session.

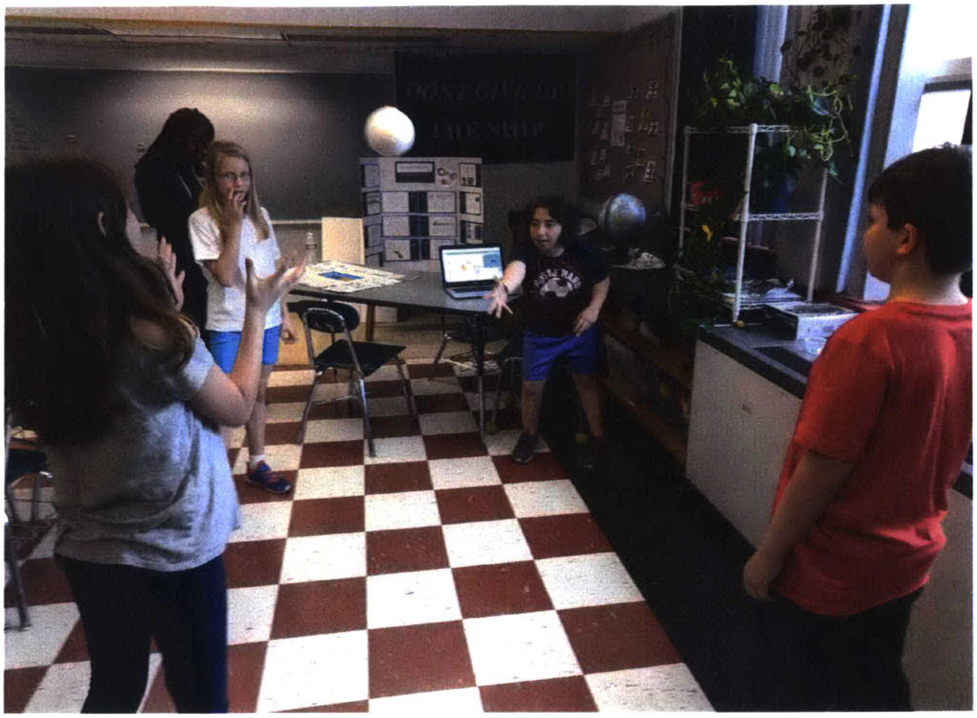
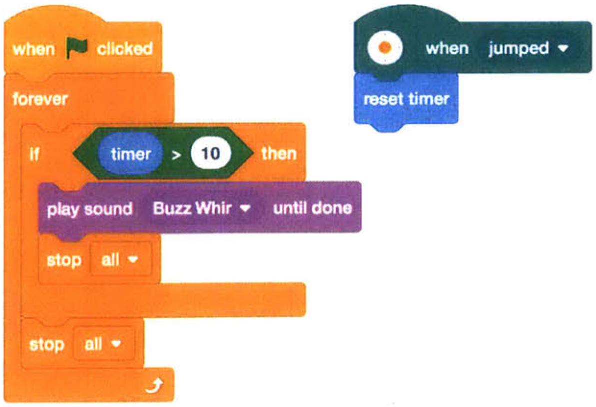


**Fig. 6.1** Suzanna's Hot Potato game

This was a major transition point in how Suzanna viewed Scratch. She was now thinking about objects in her own life and seeing how they could be applied to her explorations with Scratch and the ScratchBit. Scratch was no longer a tool to complete an assignment but instead it was something she could use to create things that she is passionate about.



```
when ⚑ clicked
forever
  if < (timer) > (10) > then
    play sound [Buzz Whir ▾] until done
    stop [all ▾]

  stop [all ▾]
↻

● when [jumped ▾]
reset timer
```

**Fig. 6.2** Main programming blocks for Suzanna's game

## Adonis, Age 15

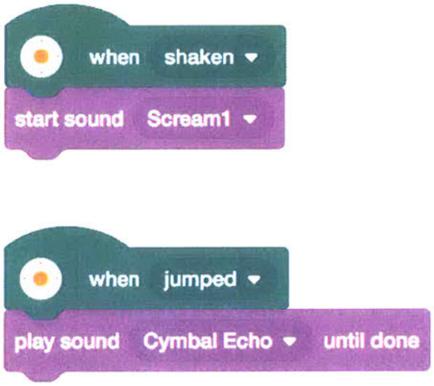
This project was one of my favorites through all of the ScratchBit play tests. This young kid came to a weekend ScratchBit play test we were holding at the South End Technology Center (SETC). He had used Scratch before and was curious about this new device we would be showing off. We introduced our demo projects and wonderful objects in the usual way, but Adonis wasn't interested in starting with physical materials. Instead, he held the device in his hand and started to code a series of responses that were recorded in his own voice. Tilting the device left would result in an affirmative response, while tilting right would result in a negative response.

Adonis remained entertained with this idea for a short time, but quickly began looking for new ideas to add to his project. I decided to remind Adonis about the objects we had brought with us, and since his project focused on a series of questions and responses I suggested he think about a designing a character for the physical part of his project.

Adonis came back with a blue teddy bear and immediately strapped the ScratchBit around its waist using rubber bands. Now as he tilted the bear around, it appeared as though the Scratch project was responding to it instead of just a controller Adonis was holding in his hand. This must have been quite the revelation because Adonis was off and running after this. He ran back to the objects and brought back a toy truck and an inflatable guitar and set to work.

The final project featured a teddy bear holding a guitar and strapped on to a toy car (Fig. 6.3). The Scratch project had a custom scream sound effect for the bear and a rock music loop that would trigger as soon as the bear hit the jump. When we all gathered around for the final share out, Adonis proudly told the backstory for the bear jump and demonstrated how it worked by thrusting the brave bear towards the ramp.

**Fig. 6.2** Adonis' Teddy Bear Jump



```
when shaken
start sound Scream1

when jumped
play sound Cymbal Echo until done
```

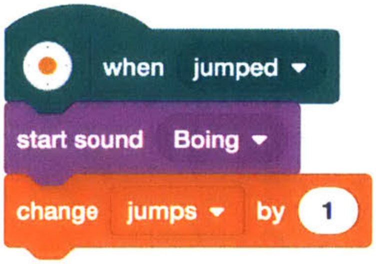
**Fig. 6.3** Programming blocks generate the Teddy Bears behavior

## Apollo, Age 7

I met Apollo by conducting a ScratchBit play test at Parts and Crafts, a local community makerspace. Apollo was full of energy. From the moment we walked through the door he was moving around, asking questions about the materials we brought with us (in particular our inflatable orange cow).

We often try to pair kids to work together during our play tests so they can explore in a collaborative way. But in this case, Apollo really wanted to work alone. He started simple with some Scratch blocks that played a "pop" sound every time he moved around. After that, he took some time walking around the room (while "popping") to observe what some of the other groups were working on.

**Fig. 6.5** Apollo jumping with the ScratchBit attached to his pocket



```
when jumped
start sound Boing
change jumps by 1
```

**Fig. 6.6** Apollo's program to count his jumps

By observing what others were working on, Apollo got inspired to add a simple counting system to his project. He created a variable labelled "jumps" and added one to it each time he jumped. Apollo spent the rest of the play test bouncing round the room looking at what others were making, while constantly circling back to his project to check his jump counter and experiment with different programming blocks. While this example may not be an excellent example of blended making, the physical creation was simply the ScratchBit attached to Apollo's pocket, I wanted to highlight this example for how it inspired the child to integrate their body movements into Scratch.

# 7. Styles of Blended Making

Throughout play testing with the ScratchBit, several different patterns, or styles, of blended making emerged. These styles describe the relationship between the physical and virtual creation. Many more styles exist, however, these are the most prominent ones that I observed throughout the play testing. The four styles are:

1. Mimicry
2. Physical-Virtual Interplay
3. Augmenting Objects
4. Body Extensions

## Mimicry

One of the most common patterns that emerged was mimicry. That is, mimicking the behavior of a physical object with a virtual analog. Some children would create the same character in both the physical and virtual world. After affixing the ScratchBit to their physical creation, they would begin to map its movements onto its virtual counterpart. Mimicry seemed to be a natural place to start if a child was unsure of how to connect the physical and virtual worlds together.

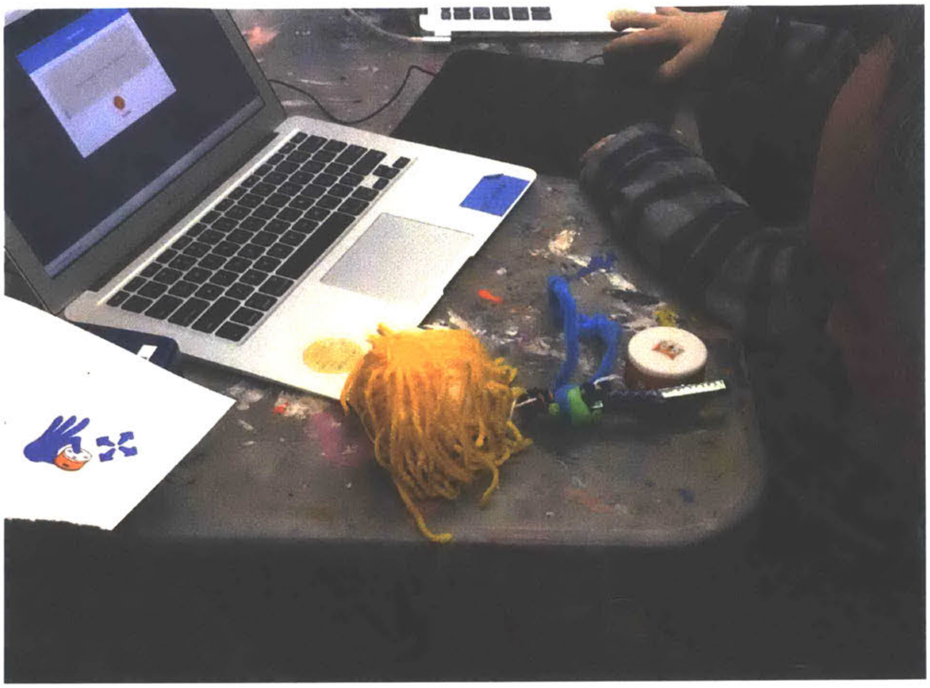


**Fig. 7.1** A custom doll made of craft materials with a virtual character mimicking its movements

In one example, a doll was constructed out of craft materials (Fig. 7.1) with the ScratchBit affixed to its midsection. Each time the doll would tilt left, a character on the screen would move left. Each time the doll was tossed into the air, the on-screen character would jump into the air. If you shook the doll, the on-screen doll would respond with a sound effect that the child had recorded.

## Physical-Virtual Interplay

Unlike mimicry, this style usually only surfaced if the individuals using the ScratchBit had a fair amount of experience using Scratch. I like to think about this style as a kind of conversation between the physical and virtual creations.

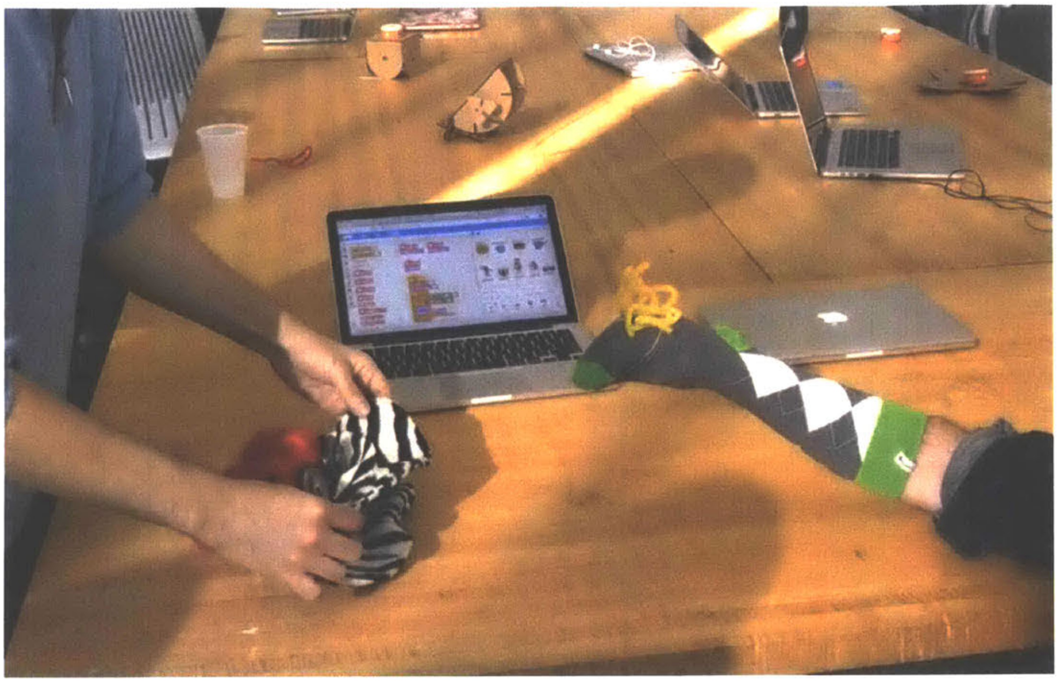


**Fig. 7.2** ScratchBit puppet show demonstrating physical-virtual interplay.

In one example, a physical puppet show was arranged with one of the puppets having a ScratchBit attached to it (Fig. 7.2). The virtual creation used various sounds to set the stage for the interaction. When the individuals performed with the puppets, both the physical and virtual interactions were generated from one another. As the puppet would move or become exposed to light, new sounds would emerge from the virtual project. This in turn would affect how the puppet actors would respond. This back-and-forth between the physical and virtual creation left plenty of room for improvisation which is why I think about this style as a type of conversation between the physical and virtual creations.

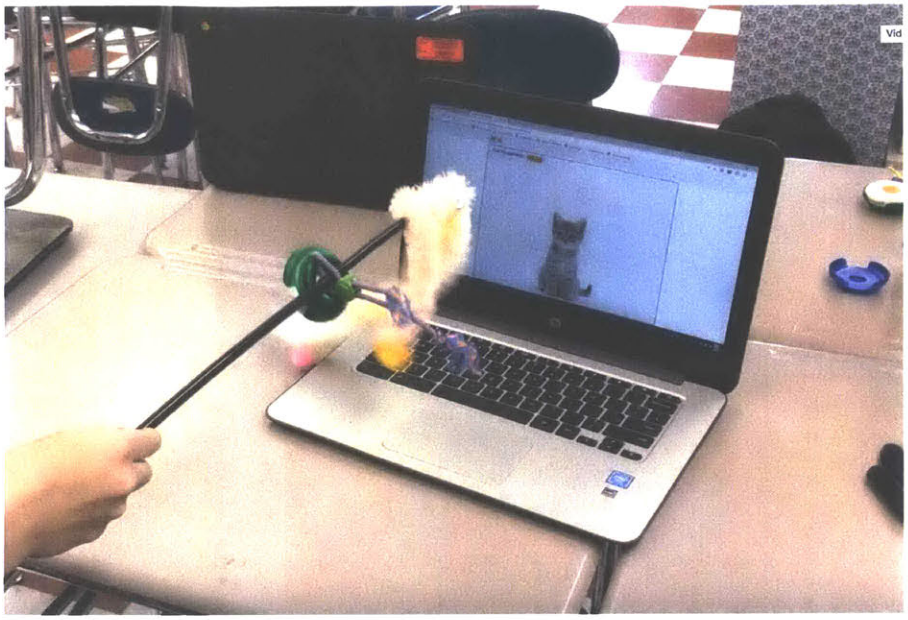


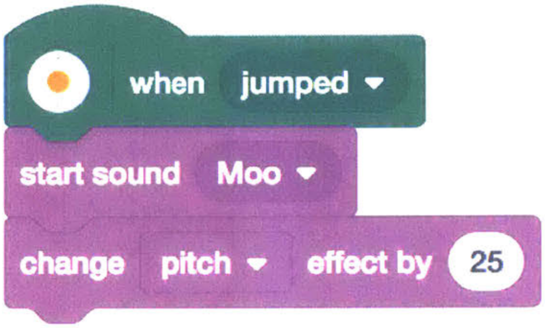
**Fig. 7.3** A virtual cat project featuring a physical cat toy interface

In another example, a physical cat toy was designed to interact with a virtual pet cat (Fig. 7.3). The virtual cat would randomly become sleepy or hungry, and the person holding the cat toy would have to complete certain gestures to satisfy the cat's needs. In this example, the individual interactions were scripted (e.g., the cat would get hungry and want you to move the toy a certain way, or shaking the toy a certain way would cause the cat to react), but the overall structure of the project would change each time someone played with it. The back-and-forth conversation taking place between the physical and virtual creations was not scripted.

## Augmenting Objects

Augmenting objects is the process of taking some physical object, either custom-made or prebuilt, and extending its functionality using a virtual creation. Paired with the wonderful objects, this is usually how a ScratchBit play test would begin: a simple challenge of choosing an object, figuring out how to attach the ScratchBit, and augmenting it by making it do something new and surprising.

One example of this style of blended making was a popular example project we would usually live code with play test participants. The ScratchBit was attached to a large, orange, inflatable cow to look like a cowbell. Then, a custom moo sound effect was recorded into Scratch. Finally the project was coded so that the moo sound effect would play each time the cow was tossed into the air. We even modified the project (Fig. 7.4) to increase the pitch of the moo sound after each toss to really convey the cow's displeasure.



```
when jumped ▾
start sound Moo ▾
change pitch ▾ effect by 25
```

**Fig. 7.4** Cow toss programming blocks

In another example, someone connected the ScratchBit to an old typewriter. Each time a key on the typewriter was pressed, a random barking sound effect would trigger from Scratch. The child exclaimed, "I made a barking typewriter" when they discovered what they had just created. The previous day, this particular learning setting was disassembling typewriters to learn how they worked. This young child saw this not only as an opportunity to understand how an existing system works, but how he could augment the system to have brand new functionality.

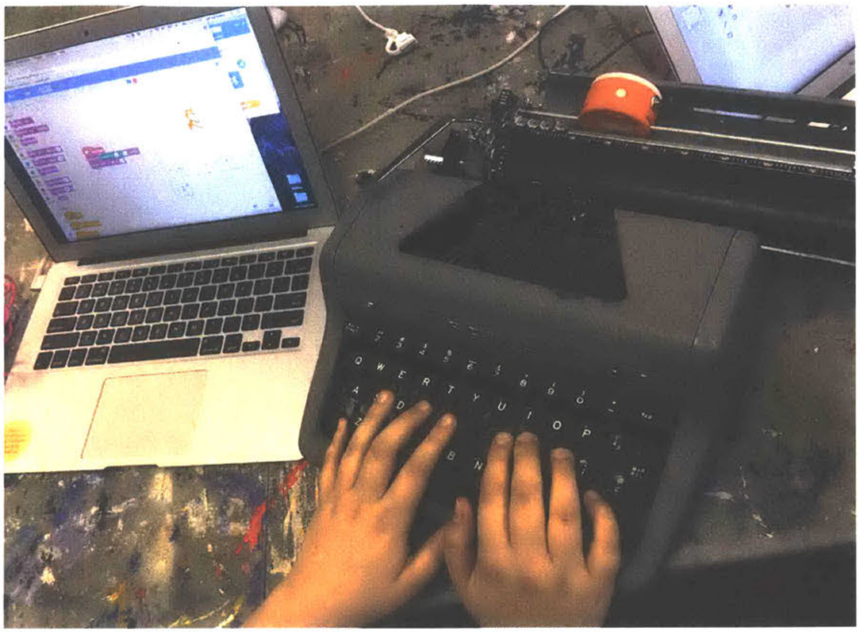


**Fig. 7.5** Barking typewriter project

## Body Extensions

The final prominent style that emerged was body extensions. As the name implies, this is done by the attaching the ScratchBit to the body, commonly clipped onto the pocket or shoelaces, or in some cases designing a physical costume that the individual could wear. With this style of blended making, body movements could be mapped to unique interactions within the virtual creation.

A simple example was Apollo's jumping project described earlier. Apollo used the ScratchBit to trigger a "boing" sound effect each time he jumped. To Apollo, he had given himself an ability that he didn't previously have, the ability to audibly "boing" as he jumped. He also added a counter in Scratch to keep track of how many times he jumped around the room. Apollo bounced himself around the room to see what others were working on all the while increasing his jump counter. When he got to 999 jumps, he called us all around to watch his one thousandth jump. Exhausted, Apollo held onto the table to keep his balance and "boinged" one last time to a cheering audience.

A more advanced example of a body extension was a musical dancing project that one child made. The ScratchBit was attached to the child's wrist via a custom wrist strap made of craft materials. As they tilted their hand in different directions, various musical loops were played to which they had different dance routines prepared. The child saw this as a way of extending their body with music. With their new ability, they weren't dancing to respond to the music but rather the flow of the music was now under their control (i.e., it was now part of their body).

# 8. Looking Ahead

## Future work in Blended Making

As I mentioned earlier in this thesis, blended making is not a novel concept. Instead, I wanted to provide a way to think about and describe a process which sometimes I see children engaging. My hope is that this will deepen the discussions around construction kits that promote both physical and virtual forms of creation.

I may wish to continue to refine the concept of blended making perhaps developing a framework for designers of construction kits for children. As continue to see tools like Scratch and Arduino be integrated into school curriculum, it will be important to be reminded of how these tools can be used for personal expression rather than just as a teaching medium for computer science concepts.

The traditional Computer Science pathway didn't work for me as I'm sure it doesn't work for many of today's children. If instead we can provide more ways for children to use technology to express themselves and explore their own ideas, we can hopefully allow more children to develop as creators and not consumers of technology.

It's my belief that blended making can provide more of these opportunities. As was demonstrated in several of the case studies earlier in this thesis, some children prefer more physical, tactile explorations. By opening up creative possibilities to both the physical and virtual world, I hope to demonstrate how we can engage more children in the joys of making.

## Bringing ScratchBit to Market

In addition to providing a wonderful platform for me to explore the ideas behind blended making, the Scratch team hopes to bring the ScratchBit to market sometime in 2019. There is still a lot of work that will need to be done around manufacturing, packaging, and distribution. But the past two years of development have given us a strong foundation to carry this work forward.

Before we move forward into manufacturing, we will have to re-evaluate our hardware design for scalability. For example, in our hardware iterations we are using a bluetooth module that comes pre-certified for radio communication globally. This module saves us significant time, but comes with a not insignificant cost difference with uncertified chips. We will have to balance out time and cost to make sure that we are able to produce the ScratchBit at a scale and cost that we deem acceptable -- again, keeping in mind our design principle of *Keep the Cost Low.*

Packaging will be another interesting area to explore in the future. Packaging can increase the overall cost to produce the ScratchBits, but it can also provide another opportunity for us to provide projects examples and getting started resources. One early idea was to be able to repurpose the packaging itself into various project templates.

Finally, distribution will be a major consideration as we will need to be able to ship the ScratchBits to Scratch users all around the world. In addition, setting up infrastructure for providing hardware support (i.e. returns/repairs) will be a major undertaking.

## Future Scratch I/O Device

Before we created our first prototype, we enjoyed thinking about what a Scratch device that has both inputs and outputs might look like. Once the ScratchBit is released, I and others on the Scratch team would like to start early explorations into this next device.

There are many ways to think about what kind of functionality an I/O device for Scratch might include. Should it be a modular design or single-board? Should it be extendable with additional sensors and actuators, or only allow preselected components? Can it be programmed and disconnected from Scratch or will it need to remain tethered to operate? These are just a few of the many questions we would consider while thinking about the next generation.

What this device looks like or how it functions are still to be determined, but it will likely follow similar design principles as the ScratchBit as these were indicative of the design philosophy of the Lifelong Kindergarten research group.

# Acknowledgements

So many people have supported me in my journey getting to the Lifelong Kindergarten group at the MIT Media Lab. There are so many people I wish to thank, that I am sure someone will have been forgotten from this list. So if you are reading this, thank you! Thank you for taking an interest in childhood creativity. Through our continued support and advocation we will hopefully one day provide ALL children with the opportunity to develop as creative explorers of this great big world.

Whenever I think about all of the incredible people who have inspired me throughout my life, I am always reminded of this quote:

"If I have seen further it is by standing on the shoulders of Giants." (Newton and Hooke, 1675)

You are all truly my giants.

To my father, John. I would not be here today if it weren't for you. Thank you for every sacrifice you have ever made for me. To my beautiful wife, Michelle. Thank you for sticking right by my side throughout this entire journey. And thank you to the rest of my family: Memere, Pepere, Keith, Debbie, Jenna, Terrell, Kaydence, Makenzi, and Brody for your never ending love.

Thank you, to the Lifelong Kindergarten group and the Scratch Team. You are so much more than colleagues. You are family. Thank you for inspiring me each and every day we spend together. A special thank you to Natalie, Andrew, Eric, Jie, and Paula for all of your help as I worked towards this thesis. To my fellow students, past, present, and honorary: Moran, Carmelo, Shruti, Stefania, Sean, Jaleesa, Lily, Marian, Yusuf, and Oliver, I value every minute we are able to work and play together. And to the extended Lifelong Kindergarten family, Amos and Liam at the LEGO IdeaStudio, all my friends at The Tinkering Studio led by Karen and Mike, Susan over at the South End Technology Center, and all my friends at Parts and Crafts.

Philipp, Katherine, and Yumiko thank you for bringing together researchers around this broad topic of learning through Media Lab Learning Initiative. I deeply cherished my time as a Learning and LEGO Papert Fellow.

To my colleagues at Lesley University, Sue, Jacy, and Anne. Thank you for pulling me into this world and for giving me seemingly endless support and respect. The work you do is crucially important and I look forward to continuing our work together.

For my thesis readers, Karen Brennan and Brian Silverman. Karen, thank you for your thoughtful feedback during my thesis writing and for electrifying my introduction to the MAS program with T550. Brian, thank you for sharing your wealth of knowledge and insights with me. You have always been extremely generous with your time and I deeply cherish our time imagining together.

And finally, for Mitch. Words will never be able to express the amount of gratitude I feel for what you have done and continue to do for me. Thank you for seeing the potential in me even when I sometimes struggled to see it in myself. Thank you for giving me your time, support, and most importantly your respect.

# References

Dougherty, D. (2012). The Maker Movement, Innovations: Technology, Governance, Globalization," vol. 7, no. 3, pp. 11–14.

Duckworth, E. R. (2006). "The having of wonderful ideas" and other essays on teaching and learning. New York: Teachers College Press.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E. (2010). The Scratch Programming Language and Environment. ACM Transactions on Computing Education, November 2010.

Martinez, S. L., & Stager, G. S. (2016). Invent to Learn: Making, Tinkering, and Engineering in the Classroom. Constructing Modern Knowledge Press.

Millner, A. (2009). Interface designs with Hook-ups in The Computer Clubhouse: Constructionism and Creativity in Youth Communities, pp.58-70. In Kafai, Y. Peppler, K., Chapman, R. (eds.). Teachers College Press. New York, NY.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. New York: Basic Books.

Resnick, M. (2007). All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten. ACM Creativity & Cognition conference, Washington DC, June 2007.

Resnick, M., & Rosenbaum, E. (2013). Designing for tinkerability. In Design, make, play: Growing the next generation of STEM innovators (pp. 163-181). Routledge New York, NY

Resnick, M., (2016). Mitchel Resnick: Designing for Wide Walls. Retrieved from https://design.blog/2016/08/25/mitchel-resnick-designing-for-wide-walls/

Rusk, N., Resnick, M., Berg, R., & Pezalla-Granlund, M. (2008). New pathways into robotics: Strategies for broadening participation. Journal of Science Education and Technology, 17(1), 59–69

Silver, J. & Rosenbaum, E. (2012). "Makey Makey: Improvising Tangible and Nature-Based User Interfaces." in proceedings of the International Conference on Tangible, Embedded and Embodied Interaction.

Fig. 2.4 Cat selfie booth made with Makey Makey (n.d.). Screenshot taken from the video *MaKey MaKey - An Invention Kit for Everyone.* Retrieved on August 20, 2018, from https://vimeo.com/279920496

Fig. 2.5 Makey Makey fruit piano interface (n.d.). Retrieved August 20, 2018, from https://labz.makeymakey.com/static/ui/partner/makeymakey/images/landing-banner-lg.jpg

Newton, I., & Robert Hooke. (1675). Isaac Newton letter to Robert Hooke, 1675 [electronic resource].