Makey Makey: Improvising Tangible and Nature-Based User Interfaces

Beginner's Mind Collective
Jay Silver
Eric Rosenbaum
MIT Media Lab
Cambridge, MA
beginners-mind@media.mit.edu

David Shaw
Intel Labs
Portland, OR
david.shaw@intel.com

Abstract
Makey Makey is a new platform for improvising tangible user interfaces. It enables people to make nature-based interfaces, it is compatible with all software, and it does not require the user to program or to assemble electronics. It is designed for a wide range of audiences, supporting ideation for experts and access for beginners. In the studio, participants will rapidly create several different user interfaces incorporating a wide variety of found objects, both physical and digital. There will also be opportunities to test out the newly created interfaces with each other, and reflect on the design of UI prototyping toolkits.

TEI 2012, February 19-22, 2012, Kingston, Ontario, Canada
ACM

Keywords
Prototyping, tangible interfaces, nature-based interfaces

ACM Classification Keywords
K.3.1 Computer Uses in Education

Introduction

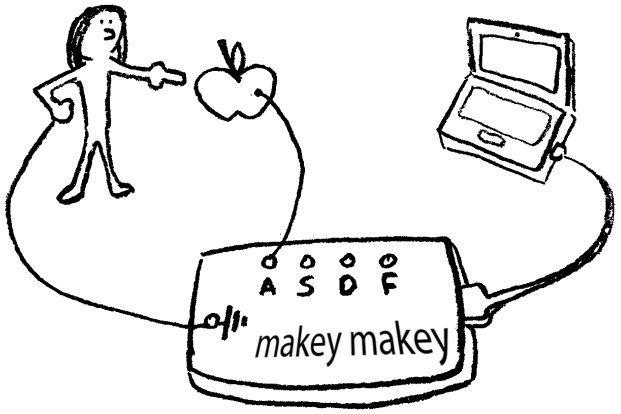

Figure 1. a Makey Makey example: the user has connected their body to ground, and an apple to the "A" key connection on the Makey Makey. By touching the apple with their finger, they will close a circuit, sending the keypress event "A" to the computer. You can imagine some software application is running on the computer, and if that software would normally make a character jump when "A" is pressed, then touching the apple would cause the character to jump (The actual device has connectors for the full set of keyboard keys and mouse events).

What if you could create a new user interface using any object, in under a minute? What if there were a new way to create tangible user interfaces that does not require skills in programming or electronics? To enable these possibilities, we propose a new platform called *Makey Makey*.

Makey Makey is a device that implements the Human Interface Device (HID) protocol, which allows it to send keyboard and mouse events to a computer without installing drivers or other software. The user connects everyday objects and natural materials to Makey Makey, in order to create a tangible user interface that controls any software running on a computer.

Our goal is to help experts ideate and to lower the threshhold for beginners to participate. We imagine some beginners are interested in creating their own computer interfaces, but do not believe they have the skills or knowledge to do so. Many experts may benefit from the ability to improvise and rapidly iterate designs. They can sketch in hardware instead of on Post-its, and quickly but coarsely explore a large design space. We aim to reach a diverse audience, including children learning through tinkering, beginners of all ages in kitchens or hacker spaces, and graduate students and professionals quickly prototyping new designs.

We see a gap in the landscape of tools for tangible UI prototyping. Our design aims to combine five characteristics of these tools that cannot be found in any single existing one:

- **Quick start for beginners**: it is possible to construct a working interface with a physical input controlling a digital output in under a minute.
- **Works with any software**: compatible with any existing computer program that normally receives input from the keyboard or mouse.
- **Nature-based interfaces**: enables creation of interfaces that include natural materials, such as plants, fruit, water, soil, or the human body.
- **Programming not required**: the system can be used to create an interface without writing a computer program.
- **Soldering not required:** complex electronics assembly such as soldering components to a circuit board (or building them on a breadboard) is not necessary.

Existing Approaches

The currently available tools for prototyping tangible user interfaces each have some but not all of these properties. Here we review three approaches: microcontrollers, sensor boards, and hacking existing devices.

Microcontrollers

Creating a tangible UI using a microcontroller is a common approach among people with technical skills. For example, the Arduino microcontroller platform can be connected to digital or analog sensors, and programmed to process its input and send data to a computer. This approach is flexible, because custom electronics enable the full range of sensing possibilities, the sensor data can be conditioned as needed, and it is typically used with custom software receiving the sensor data.

Sensor Boards

An easier but less flexible approach is to use custom sensor boards that work with customized software. For example, the Scratch Sensor Board is designed to be used with the Scratch programming environment, a graphical language for creating interactive multimedia projects. It

enables a very quick start with little setup, but it can only be used with the Scratch Software.

Hacking Existing Devices
A third approach is to modify typical computer user interfaces such as a mouse or keyboard. For example, one trend in the maker community is to run workshops on hacking computer keyboards, adding your own connections that act as switches and send keyboard events to the computer. Breaking open the keyboard and modifying the printed circuit board in order to solder to it is challenging for beginners and error-prone. This approach has the advantage that no programming or custom software is required— it works with existing software that is controlled by the keyboard.

	Micro-control-lers	Sensor Boards	Hacking Existing Devices	Makey Makey
Quick start for beginners		Yes		Yes
Works with any software			Yes	Yes
Nature-based interfaces	Yes			yes
Programming not required			Yes	Yes
Soldering not required		Yes		Yes

Table 1. Characteristics of different tools for prototyping tangible user interfaces.

The Story of Makey Makey

The inspiration for Makey Makey comes from two places: our own needs, and the generalization (or "platformization") of a previous project.

Sometimes, when you want to quickly create an interface, the easiest way is to hack a keyboard or mouse, rather than using a microcontroller. We have found this in our own work, and observed others doing it as well, such as museum exhibit designers. There seemed to be a niche for a tool that made this type of hack even easier.

The other source of inspiration was a project we have collaborated on for several years that, through its blossoming, suggested a manifestation as a new platform. Drawdio is a toolkit that we designed for creating musical instruments out of everyday materials by completing a circuit, generating a sound with a frequency proportional to the resistance. Its most common manifestation is as a pencil, with which you can draw a musical instrument and then play it, but it can also be used as a toolkit to create other instruments, using things like human bodies, food, plants, and the kitchen sink. Drawdio is simple and accessible enough that it makes designing interfaces possible for a wide range of people, but it is constrained to generating musical tones. We imagined a platform that was just as accessible but enabled people to make a much wider variety of interfaces. This vision was combined with the idea of keyboard hacking (or keyboard emulation), and Makey Makey was born.

Nature-Based Interfaces

Part of the magic of Makey Makey is that it is not limited to closing switches by closing circuits in the usual way, reducing the resistance nearly to zero ohms. Instead, it is designed to respond to a resistance dropping below a

threshold in the range of tens of megaohms. This enables a circuit to be closed by a connection through the human body, or a wide range of natural materials such as food, plants, and soil. This design was inspired by the wide range of creative uses we have seen people make of the Drawdio kit, and we hope that Makey Makey will similarly enable people to explore the creation of interfaces that incorporate a wide variety of natural materials.

Combining Physical and Digital Bricolage

The flexibility of Makey Makey for creating physical interfaces out of everyday objects, especially including natural materials, means it will facilitate bricolage in the physical world. At the same time, because it is compatible with just about any software you can think of (anything that takes keyboard or mouse input), it can be used for a kind of digital bricolage: Want a specific application? Google it instead of making it yourself. In this way, we believe that Makey Makey as a platform opens up a new space for bricolage, combining creative use of found materials in both the physical and digital worlds.

Studio Proposal

We propose a workshop in which participants will quickly create a series of different user interfaces using Makey Makey.

We aim for Makey Makey to be as accessible as possible, so the requirements for the workshop are minimal. We require nothing but a desire to make a UI or interactive experience.

Our studio is relevant because these types of topics come up over and over again in TEI, and this workshop will give people an experience of having co-created many hands-on examples which they can draw on to think with in future UI design.

Learning Goals / Topics Covered

- Nature-Based User Iterfaces
- Human skin interfaces
    - Plant interfaces
    - Water interfaces
    - Edible interfaces
    - Pencil/Paintbrush-as-interface
- Ultra fast iterative prototyping (< 10 minutes per UI)
- Physical and Digital Bricolage
    - Computer Crafting
    - Resistance Sensing (optional)

Each person will have completed 4 new UI's by the end of the workshop, two simple ones and two more complex ones. Each person will see a wide range of examples build by their peers and demonstrated or shown by the facilitators, all of which helps participants imagine the large space of tangible user interfaces that can be quickly prototyped with Makey Makey.

Workshop Outline

Phase 1: Warmup / Intro
Separate ourselves from the rules of society, get into a creative space, and introduce the topic.

Phase 2: Lightning
Fast iterative prototyping rounds paired with show and tell.

Phase 3: Make
Two deep building phases seperated by a shuffle and capped by a user experience test.

Phase 4: Reflection
Discuss with respect to our own processes, the educational value of design, and design thinking.