Rapid Prototyping of Tangibles with a Capacitive Mouse

Juan David Hincapié-Ramos
IT University of Copenhagen
Copenhagen, Denmark
jdhr@itu.dk

Morten Esbensen
IT University of Copenhagen
Copenhagen, Denmark
mortenq@itu.dk

Magdalena Kogutowska
IT University of Copenhagen
Copenhagen, Denmark
magk@itu.dk

ABSTRACT
This paper presents the Toki toolkit: a do-it-yourself guide and API to support the rapid prototyping of tangibles. The toolkit provides support for two common requirements for tangibles: capture of touch input by an user and communication of such input to a computer. At the core of the toolkit lays the capacitive surface and communication capabilities of a Microsoft TouchMouse, both of which are appropriated to fulfill the mentined requirements. Unlike existing approaches for rapid prototyping of tangibles like the Arduino boards, using the Toki toolkit does not require developers/designers to program a chipset, configure wireless interfaces, and define and implement communication protocols. The do-it-yourself guide illustrates how to create a cover for the mouse required to re-use its capabilities. The API offers a set of services to develop computer applications that interface with the tangible.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces: Input Devices and Strategies, Prototyping

General Terms
Design

Keywords
Tangibles, D.I.Y, Rapid Prototyping, Framework

1. INTRODUCTION
Tangible interaction uses electronically augmented physical artifacts to trigger digital events. Its implementations cover a wide field of interdisciplinary approaches ranging from playful installations supporting learning [10, 11, 12], to practical tools for interacting with digital information [6]. The core feature of tangible interfaces is the coupling of physical artifacts to computational models [13] and/or digital information. In most cases, the tangible receives user input, computes it according to a predefined model, and provides feedback which can be visual, auditory or haptic.

Figure 1: The touch mouse and the Toki pyjama.

In designing tangible interactions, designers usually follow an iterative process. Not too long ago, this was an expensive and time consuming process that required both programming and engineering skills. More recently, new technologies help cut both time and costs for *rapid* prototyping, as well as lower the programming and electronic hacking skills needed. An example to draw upon is the Arduino platform[1], a tool specifically designed to support the rapid prototyping of hardware by abstracting the complexity of circuit design and low level programming. This simple platform is aimed at learners and hobbyists.

However, despite these efforts to make the field of tangible interaction more accessible, much work remains to be done. For example, in order to build a simple tangible wireless controller for a computer application, even the Arduino platform poses a high level of complexity, requiring to: 1)- wire the artifact (optionally solder components), 2)- program the Arduino board, 3)- add a wireless communications shield to communicate with the computer, and 4)- define and implement a communications protocol between the two. On the other hand, the input capture and communication capabilities already exist in most computer peripherals like the mouse, which can communicate relative movement (x,y) and different click events. A special case is Microsoft's capacitive Touch Mouse[2] which communicates the state of its active capacitive surface as a byte matrix (13x15).

This paper presents the **Toki toolkit**, a method to appropriate the input handling and communication capabilities of the Microsoft Touch Mouse and use them to support the rapid prototyping of tangible controllers without the need to program the tangible object. The toolkit relies on a custom made cloth cover – pyjamas – (see figure 1) which tightly

[1]http://www.arduino.cc/
[2]http://goo.gl/d7bPJ

wraps the mouse and offers multiple contact points that can be linked to different parts of the tangible by means of metallic cables or conductive thread. The kit is composed of by a do-it-yourself guide to create the pyjamas and a software API to handle the events happening at the contact points.

2. RELATED WORK

At a conceptual level, tangible user interface (TUI) researchers have suggested a variety of taxonomical frameworks, concerned with defining terms, categorizing and characterizing systems, and types of coupling [5, 8, 13].

At a practical level, multiple projects studied development support for tangible interaction; consisting mainly on detecting interactions with physical objects and associating these real world events with a virtual counter part. Papier-Mâché [7] uses multiple sensing techniques (RFID, image recognition, and barcodes) to recognize objects and allows users to associate custom routines to the appearance of disappearance of such objects from the display. Phidgets [4] and iStuff [1] take a different approach and provide a set of pre-determined hardware components (sensors and actuators) which are supported by the framework and can be programmed by the developer. Hardware platforms like Arduino integrate with a wide range of standard electronic components requiring circuit building, wire soldering and programming in the *processing* language. The LilyPad Arduino [2] is designed for wearables and e-textiles, and therefore can be sewn to fabric and connect to power supplies, sensors, and actuators with conductive thread.

Moreover, educational computing research studies tangibles through the notion of *digital manipulatives* – computationally-enhanced versions of traditional children's toys [11] to support learning. A large subset of these educational toys is modular such as Topodo [10], LEGO Mindstorms [9], and Spelling Bee[3]. This modularity encourages creative thinking as each 'bit' can be put together with another 'bit' in new and unexpected ways, which suggest emergent affordances.

Our toolkit design aims to simplify the physical prototyping of tangibles. And although creating tangible interactive user interfaces is still non-trivial, there now exists technologies, like capacitive touch surfaces (phones/mice), which can be reappropriated to build simple forms of tangible interaction.

3. TECHNICAL APPROACH

The Microsoft Touch Mouse is a wireless mouse with a capacitive surface capable of detecting contact with a *grounded* object; the human skin is recognized as *ground* and this is the base for touch interaction. The mouse surface is represented as a 13x15 matrix with individual byte values (0-255) indicating touches at the each point of the surface. The mouse communicates this matrix to the computer it's paired to at a 120fps rate. The Microsoft Touch Mouse provides an API that receives this matrix and passes it on to any software application. Figure 2 shows the user touching the mouse and its matrix representation.

Due to the nature of capacitive surfaces, it is possible to extend the surface by adding wires or other conductive materials. Touching the wires will still change the capacitance of the surface and the change is reflected on the matrix. Our

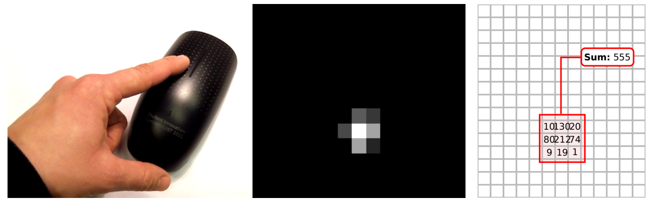

Figure 2: Interaction representation.

proposal is to use this extensibility affordance of the mouse to wire different parts of a tangible to pre-determined areas of the mouse surface. Figure 3 shows how the capacivite surface registers different values when the user is touching or not, and when the touch is directly on the surface or through a conductive wire extension.

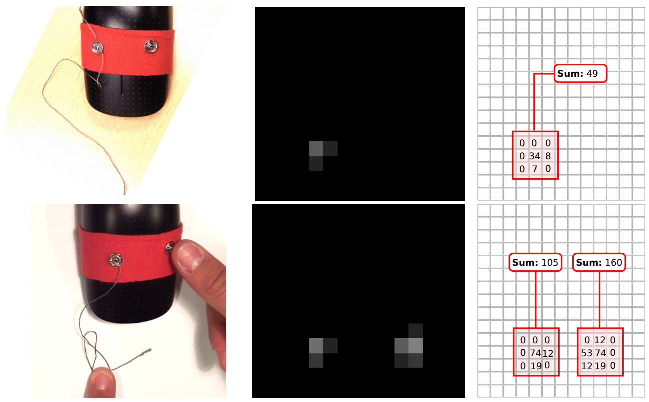

Figure 3: Interaction with and without extension.

Figure 3-top shows a challenge arising from extending the surface through conductive wires: the wire itself is registered on the mouse surface even though it's not being touched by a person. Figure 4-left shows the sum of all contact points for two different zones (a zone is a 3x3 square in the mouse matrix) where one is not extended and the other is extended with a 20cm conductive thread. The average sum of values for the zone once the wiring is added is what we call the **zone baseline**. The change on the sum when a human touches the wiring is what we call the **target margin**. Figure 4-right shows how the length of the wiring on each zone impacts both the zone baseline and the target margin. The longer the cable the higher the zone baseline and the smaller the target margin.

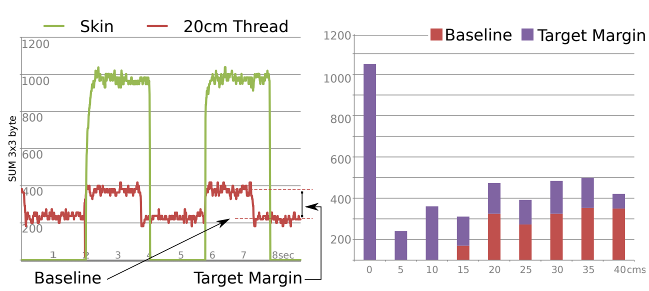

Figure 4: *left* – baseline and target margin on surface and at 20cms; *right* – target margins at different lenghts.

The Toki D.I.Y. toolkit aims at supporting amateurs in building and experimenting with simple designs of tangible interactions. The toolkit leverages the extensibility affordance of capacitive surfaces in the construction of tangibles, in a way that does not require prior knowledge about electronics, and consequently has the potential to alleviate some of the initial difficulties that novices to the field can experience. The toolkit's central element is a mouse pyjamas: a cloth wrapper for the mouse that holds 9 metal buttons in a 3x3 array, each touching the capacitive surface of the mouse. As the buttons are conductive, touching one will be registered as a touch on the mouse at the point where the button is located. By adding wires or conductive thread to each of the buttons, the surface is further extended and will register touch when the naked wire is touched. This construction is the core of the Toki toolkit - by wrapping the mouse in the pyjamas and adding wiring between this and any object, the mouse will recognize touches at the end of the wiring. The mouse with the pyjamas can then be added to any physical object to augment it with touch-recognizing capabilities.

In order to build and leverage the mouse pyjamas, the toolkit is made of two components which are explained in the following sections: 1) a do-it-yourself guide, and 2) a .NET C# software API and library.

4. D.I.Y GUIDE

The do-it-yourself guide presents a set of steps required to make the mouse pyjamas. The guide is a simple illustrated step-by-step brochure intended to be easy understandable. The guide takes the reader through the following steps: 1) cut out paper templates, 2) draw the templates on the cloth, 3) cut out the shapes of cloth, 4) sew together the pieces of cloth and add the velcro, 5) snap on the buttons, and 6) insert the mouse. Image 5 shows step 3 from the guide: cutting out the two pieces of cloth that make up the pyjamas body.

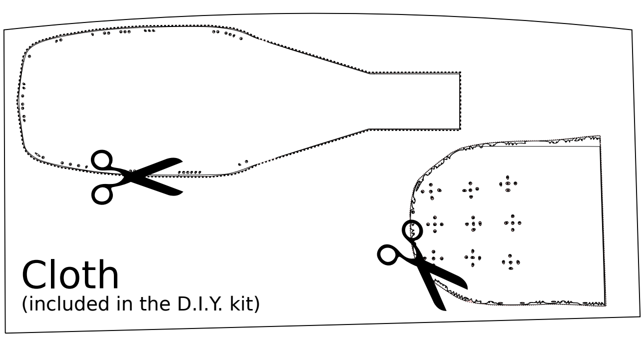

Figure 5: Example step from the DIY guide

As an example, the do-it-yourself guide shows how to make an interactive stuffed toy elephant by using our prototype out-of-the-box toolkit. In addition to a Microsoft Touch Mouse, the toolkit contains the following items: cloth, 9 snap buttons, needle, velcro tape, manual (includes paper pattern), normal and conductive thread. The guide, API, and sample application can be downloaded from the Toki website: http://itu.dk/people/mortenq/loki/.

5. API

The second component of the toolkit is an API that hooks into the mouse and translates the matrix input to a set of events on the 9 predefined zones. Moreover, the API handles calibration and provides mechanisms for recording and playback of interactions with the mouse.

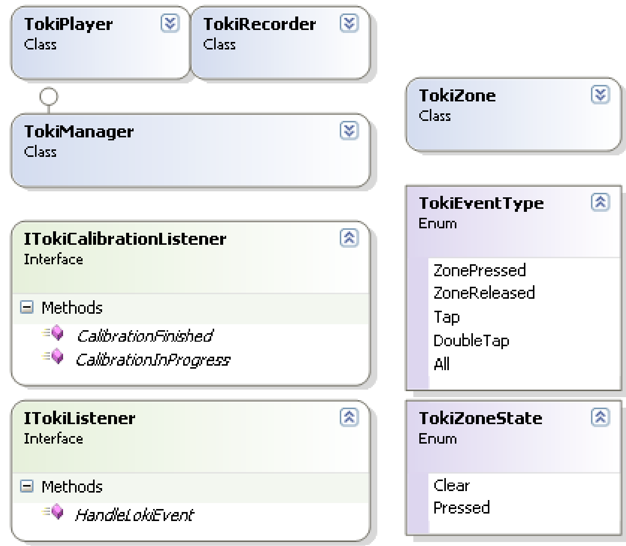

Figure 6: Main components of the Toki API.

Figure 6 shows the main components of the Toki API. The central component is the *TokiManager* class to which the developer registers an object implementing the *ITokiListener* interface. The API is capable of recognizing 4 different *TokiEventType* events from each *TokiZone*: 1) finger down, 2) finger up, 3) tap, and 4) double tap. When any of the events is recognized by the TokiManager the application gets notified via the ITokiListener object. The API will only notify the programmer of events that she explicitly subscribed to.

As mentioned before, the mouse will recognize input signals when wiring is added even though no touches occur. To compensate for this, the API offers a calibration mechanism. Calibration should be performed each time a Toki application is started to ensure maximum precision of the touch detection algorithms. The calibration process is started by inserting the mouse into the pyjamas and invoking the calibration method of the API. The calibration algorithm will detect the average noise at each zone as well as the standard deviation. Once the calibration has finished, these values are saved and now used by the touch detection algorithms. The progress and finishing of the calibration process are notified to the application through the *ITokiCalibrationListener* interface. Signals are now only treated as touches if they exceed the average noise value plus two standard deviations.

Finally, the API provides a mechanism to facilitate the testing of the interactive application: the *TokiRecorder* and the *TokiPlayer*. By using this two objects the application developer can record into a file a series of user interactions with the tangible artifact, and then replay them within the application logic, avoiding the need to constantly manipulate the object for testing.

6. SAMPLE APPLICATION

We tested the toolkit by building an elephant toy and a simple application based around the childrens' game known

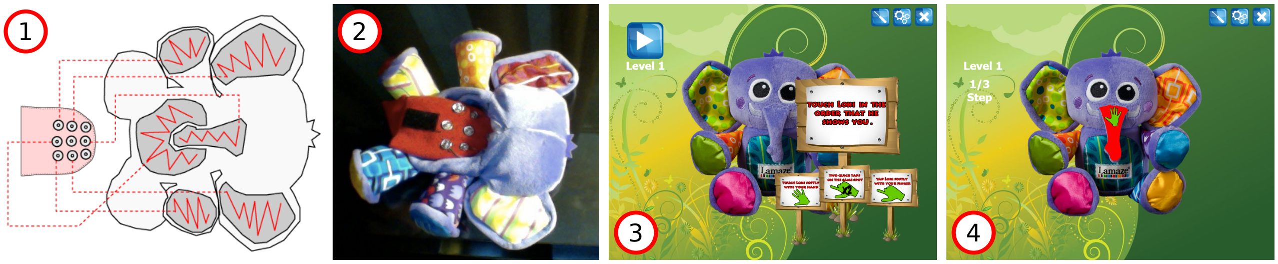

Figure 7: The Toki sample application with the elephant puppet and how it is wired to the mouse.

as ‘Simon says’. In the example application we have made 6 touchable zones on the elephant: trunk, belly, ears and fore pads; each zone is connected by the conductive thread to the mouse. An image of the toy is shown on the screen, along with instructions on how to play the game: Touch the elephant in the order indicated by the red areas on the image, the green hand icon ( ) means pat gently on the designated area, a finger icon ( ) means tap the area – touch for less than 500ms, and tapx2 icon ( ) means tap the area twice in quick succession. At the start of a game, the player is presented with a sequence of red-colored areas on the image, and is expected to remember the sequence and touch the elephant on the indicated areas using either the patting or the tapping techniques. As the player advances through the levels, they are required to remember longer sequences. Figure 7 shows 1) how the toy is wired to the mouse pyjama, 2) how the mouse+pyjama are hosted inside the toy, and 3-4) screen captures.

7. DISCUSSION

The Toki toolkit provides designers and developers with an easy toolkit for rapid prototyping of tangibles, however it has its limitations. Attaching wire or conductive thread to the mouse will generate noise. While we implemented calibration methods to handle this there’s an upper limit as to how long wiring can be. We tested our implementation with conductive thread and found this limit to be around 35 cm, however this may change with the material used. We have limited the number of contact points of the pyjamas to 9, because although it would be possible to add more zones, we expect users to make hand-made pyjamas, and so in order to ensure correct touch-recognition.

In the future we would like to look at other everyday capacitive technologies in their potential for rapid prototyping. Several smart-phones come equipped with capacitive screens and a wide variety of wireless communication capabilities; It seems possible to use these as the center of a DIY prototyping toolkit. Even though our toolkit provides a detailed guide to assemble and program a tangible prototype with Toki, it still requires the developer / programmer to be familiar with the C# programming language. We would like to look in to the possibility of adding visual programming support like for e.g. Scratch[3].

Finally we would like to perform empirical evaluations of the Toki toolkit. The first would be to determine the robustness of the toolkit. This evaluation would include testing it in different applications and with different wiring. The second evaluation should include designer and programmers to evaluate the usefulness and usability of Toki API.

[3]http://scratch.mit.edu/

8. ACKNOWLEDGMENTS

This work was presented at the UIST’11 student competition, receiving the award as best implementation. This research has been funded by the Danish Agency for Science, Technology, and Innovation partly under the project “PC Mini-Grids for Prediction of Viral RNA Structure and Evolution”, #09-061856; and under the project for Next Generation Technologies for Global Software Development. We thank Leo Sicard, Sebastian Buttrich, Afsaneh Doryab, Joel Andres Granados and Jakob Bardram for their support and multiple input to our original idea; and specially Aurelien Tabard for pushing us to work on something really worth of our time.

9. REFERENCES

[1] R. Ballagas, M. Ringel, M. Stone, and J. Borchers. istuff: A physical user interface toolkit for ubiquitous computing environments. 2003.

[2] L. Buechley, M. Eisenberg, J. Catchen, and A. Crockett. The lilypad arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proc. CHI ’08*. ACM Press.

[3] A. Dekel, G. Yavne, E. Ben-Tov, and Y. Roschak. The spelling bee: an augmented physical block system that knows how to spell. In *Proc. ACE’07*. ACM Press.

[4] S. Greenberg and C. Fitchett. Phidgets: easy development of physical interfaces through physical widgets. In *Proc. UIST’01*. ACM Press.

[5] E. Hornecker and J. Buur. Getting a grip on tangible interaction: a framework on physical space and social interaction. In *Proc. CHI’06*. ACM Press.

[6] S. Hunter, P. Maes, S. Scott, and H. Kaufman. Memtable: an integrated system for capture and recall of shared histories in group workspaces. In *Proc. CHI’11*, CHI ’11. ACM Press.

[7] S. R. Klemmer, J. Li, J. Lin, and J. A. Landay. Papier-mache: toolkit support for tangible input. In *Proc. CHI’04*, New York, NY, USA. ACM Press.

[8] B. Koleva, S. Benford, K. H. Ng, and T. Rodden. A framework for tangible user interfaces. *Interfaces*, 2003.

[9] F. Martin, B. Mikhak, M. Resnick, B. Silverman, and R. Berg. To mindstorms and beyond: Evolution of a construction kit for magical machines. pages 9–33. Morgan Kaufmann, 2000.

[10] H. S. Raffle, A. J. Parkes, and H. Ishii. Topobo: a constructive assembly system with kinetic memory. In *Proc. CHI’04*. ACM Press.

[11] M. Resnick, F. Martin, R. Berg, R. Borovoy, V. Colella, K. Kramer, and B. Silverman. Digital manipulatives: new toys to think with. In *Proc. CHI’98*. ACM Press/Addison-Wesley Publishing Co.

[12] G. Revelle, O. Zuckerman, A. Druin, and M. Bolas. Tangible user interfaces for children. *Proc. EA CHI’05*.

[13] B. Ullmer and H. Ishii. Emerging frameworks for tangible user interfaces. *IBM Syst. J.*, 39:915–931, July 2000.