Chair for Computer Science 10 (Media Computing and Human-Computer Interaction)

RWTH AACHEN UNIVERSITY

moIT: Multi-Modal-Tangible to Enhance Interaction with Capacitive Touchscreens

Bachelor's Thesis
submitted to the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Armin Mokhtarian

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr.-Ing. Stefan Kowalewski

Registration date: 14.09.2016
Submission date: 29.09.2016

Eidesstattliche Versicherung

______________________________ ______________________________

Name, Vorname Matrikelnummer

Ich versichere hiermit an Eides Statt, dass ich die vorliegende Arbeit/Bachelorarbeit/ Masterarbeit* mit dem Titel

____

____

____

selbständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

______________________________ ______________________________

Ort, Datum Unterschrift

*Nichtzutreffendes bitte streichen

Belehrung:

§ 156 StGB: Falsche Versicherung an Eides Statt

Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt

(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.

(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:

______________________________ ______________________________

Ort, Datum Unterschrift

Contents

List of Figures

List of Tables

Abstract

Human-Computer Interaction has expanded through the years. The introduction of tangibles allowed using physical objects as an alternate input option for touchscreens. One fundamental advantage of those is, that their function can be illustrated by their representation, to provide an exemplified interaction. Currently, one tangible is associated with one functionality, because changing the functionality means changing its physical representation, which is usually not possible during runtime.

This bachelor thesis presents a new kind of tangibles, which uses further input and output modules to provide an enhanced representation and interaction. Thus, it is for example possible to see and change the current representation (and functionality) using a touchscreen. These and other enhancements will be implemented and evaluated in the following. Moreover, this thesis presents a prototype for such an enhanced tangible.

Überblick

Die Mensch-Maschine Kommunikation hat sich über die Jahre stets erweitert. Die Einführung von Tangibles ermöglichte dem Benutzer physikalische Objekte als alternative Eingabemöglichkeit für Touchscreens zu verwenden. Diese besitzen den Vorteil, dass sie ihre Funktion mit ihrer physikalischen Repräsentation andeuten und somit eine anschaulichere Kommunikation erlauben. Momentan ist einem Tangible eine Funktion zugeordnet. Falls eine Funktion des Tangible gändert werden soll, muss folglich auch die Repräsentation geändert werden. Dies ist jedoch in der Regel zur Laufzeit nicht möglich.

Diese Bachelorarbeit stellt eine neue Art Tangible vor, die mit Hilfe von zusätzlichen Ein- und Ausgabemodulen eine verbesserte Kommunikation ermöglicht. So ist es beispielsweise jederzeit möglich, mit einem Touchscreen, die aktuelle Repräsentation bzw. Funktion des Tangibles anzuzeigen oder auch zu ändern. Diese und weitere Erweiterungen werden im Folgenden implementiert und evaluiert. Darüber hinaus stellt die Arbeit einen Prototypen des oben genannten erweiterten Tangible vor.

Acknowledgements

I would like to thank Prof. Dr. Borchers for supervising my thesis and Prof. Dr.-Ing. Stefan Kowalewski for being my second examiner.

I am very thankful for Christian Cherek who helped with constructive feedback during my work which helped me improve this thesis.

I thank my family for supporting and enabling me to write this thesis. I would like to thank Jan Thar for helping me with the electronic design. Last but not least I am very thankful for everyone at i10, providing help and great ideas which helped me during the writing of this thesis. Thank you for providing such a pleasant environment.

Also a huge thanks to everyone who proofread my work!

Thanks for all the support!

Conventions

Throughout this thesis we use the following conventions.

The thesis is written in American English.

molT is short for Multi-Modal-Tangible, which is the work name of the described tangible.

The thesis is written in first person plural. This was not used because several persons worked on this thesis but for esthetical reasons.

molT is based on a prototype created by Voelker et al. [2015]. The prototype was revisited, redesigned and extended.

Any term that is introduced for the first time is written in italic. Following appearances will not be italic. The exclusion of this convention are different programming language methods, which are always written italic.

Chapter 1

Introduction

Figure 1.1: Classic tangibles

Speaking of classic *tangibles*, we usually talk about single modal tangibles whose only function is to give tangible (physical and graspable) external representation to the digital information. Tangibles provide a beneficial way of interaction, because however powerful, *Graphical User Interfaces* are inconsistent with our interactions with the rest of the

physical representation for digital information physical world. Tangible design expands the affordances of physical objects in order to support direct engagement with the digital world.

Figure 1.1 shows a variety of tangibles built at the Chair i10 at the RWTH Aachen. Each of them gives a vivid representation of their digital information. Tangible interfaces take advantage of our haptic sense and our *peripheral attention* to make information directly manipulable and intuitively perceptible through our *foreground and peripheral senses* . Thus, the tangible representation helps bridge the boundary between the physical and digital worlds. But while the digital information can change permanently the physical representation stays static. For example, selecting the representation as a magnet makes it impossible to convert it during the runtime to anything else. That circumstance leads to the model for *tangible user interfaces* shown in Figure 1.2.

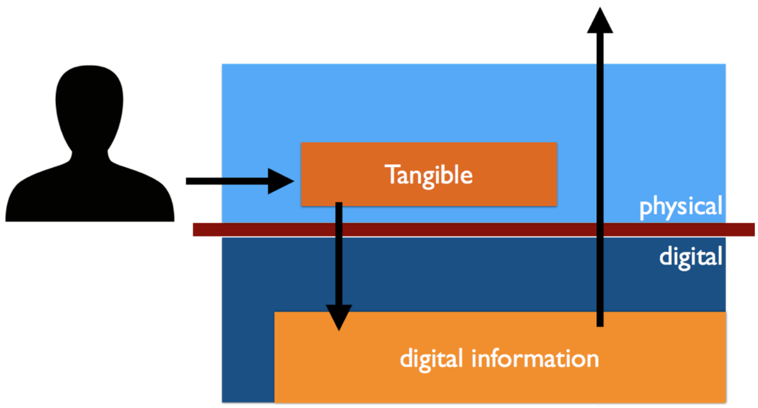

Figure 1.2: Classic tangible user interface

This model based on Ishii [2008] shows the interaction with a tangible. Although the tangible is used to interact with the system, it only works in one direction. The tangible may be used to play air hockey, but the user neither gets haptic feedback if they hit a puc, nor hear the sound of the collision, at least not from the tangible itself. To enable an extended communication, a new layer must be added to the system.

This layer is named *control layer* and was investigated by Prof. Ishii (displayed in Figure 1.3). The modification per-

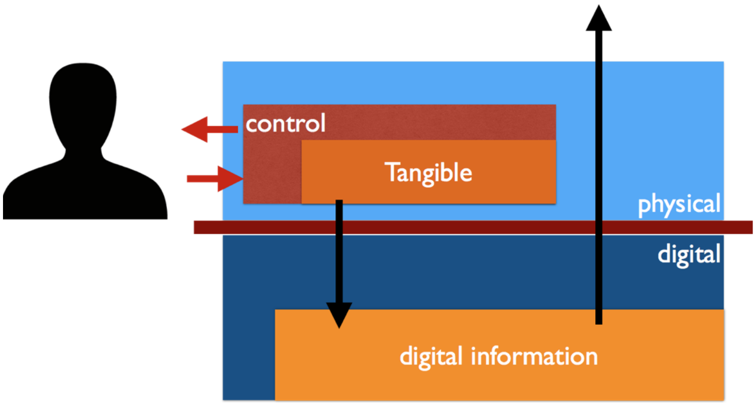

Figure 1.3: Extended tangible user interface

sist of the addition of multiple possible input and output methods to the tangible. Our goal is to build, based on human senses, a tangible which provides methods for the visual, acoustic, haptic and motion senses. In spite of its richer interaction it still should be mostly intuitive usable. Table 1.1 displays three possible input and three possible output methods. This thesis presents how to process these into a tangible and gives examples for their use. Since having multiple interfaces enables new possibilities regarding interaction with a system, we will also give an example and present a user study to evaluate the concept, based on this example.

	visual	acoustic	haptic	motion
input	-	Microphone	Touchables	Acceleration sensor
output	Display	Speaker	Vibration	-

Table 1.1: Installed interfaces

Chapter 2

Related work

Most of the commercially available multi-touch devices use capacitive touch technology. The major challenge of tangible design, which is enabling a reliable recognition, is at once the reason why tangibles did not reached everyday life. Other approaches require the user touching the tangible (Yu et al. [2011]), use modified touch displays (Liang et al. [2014]), or use touch surfaces which detect touches by infrared light (Schöning et al. [2009]). Thus, tangibles on capacitive touch technology were already researched for over a decade (Rekimoto [2002]), but *PUCs* - Passive Untouched Capacitive Widgets by Voelker et al. [2013] are the first tangibles that could be detected on commercially available and unmodified multi-touch displays.

Tangibles are hard to detect on capacitive touch

To increase the number of possible applications for tangibles on unmodified multi-touch surfaces, a problem had to be solved which arise when using PUCs. In case the PUC is not moved, the recognition will last (depending on the filter mechanics of the hardware) only about 10-30 seconds. Hence, the system can not distinguish between a tangible that was lifted and one that was filtered.

Tangibles were improved in their abiltiys

In this thesis, we revisit the approach of Voelker et al. [2015] called *PERCs*, which is the improvement of PUCs. This approach provides a reliable and long-term detection for the majority of *capacitive displays* without further modifications on the display itself. Figure 2.1 illustrates the individual

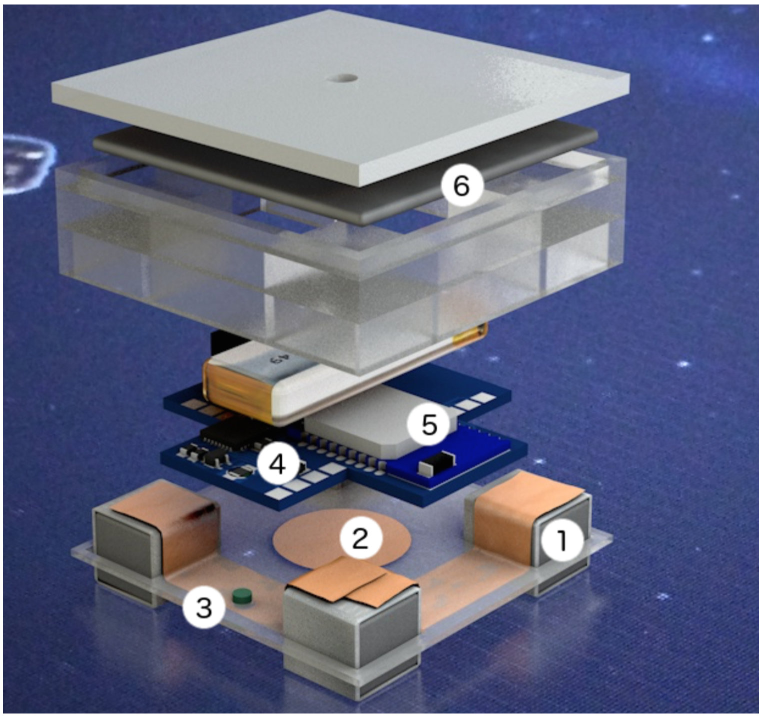

Figure 2.1: Explode view of a PERC tangible.
Image taken from Voelker et al. [2015]

PERCs provide robust and reliable detection

components of the PERC tangible. A touch pattern with at least three touch marker (1) is connected with copper. The field sensor (2) and a light sensor (3) ensure, that the tangible gets detected by the touchscreen for preferable unlimited time. Since the sensor information is transferred via Bluetooth (5), the unique Bluetooth ID can be used to identify every tangible. Furthermore, the microcontroller (4), battery, lead plate (6) and the acrylic frame are presented in the Figure. Overall, PERCs have a solid and expandable foundation, which we will use in this thesis.

Tangible design gaining importance in HCI

The relevance and necessity of tangible interaction in Human-Computer Interaction is emphasized by Hornecker and Buur [2006]. Further, they describe, that the usual challenge of designing is extended: 'Designing tangible interfaces requires not only designing the digital but also the physical, and their interrelations within hybrid ensembles, as well as designing new types of interaction that can be characterized as full body, haptic, and spatial.' This thesis will focus on the physical and their interrelations, and give an example approach for a tangible which takes this statement in mind.

Existing approach for changeable representations

Nevertheless, there already exists an approach to dissolve this lack of unchangeable representations. Ishii et al. [2012] found a way to manipulate physical forms during runtime. By snapping together a combination of passive (static) and active (motorized) components, people can assemble dynamic biomorphic forms. However, their main weakness still needs to be improved. Although they are transformable, they are limited in their possible representations and thus do not really represent everyday objects. Therefore, they would not come into question for many use cases. Bakker and Hollemans [2007] writes about the benefits of tangible interaction in a tabletop game, where tangibles represents the playing pieces. Regarding this kind of use cases, it is desirable to have a tangible with a higher range of representations.

Chapter 3

molT

As described in chapter 1, we want to design a tangible having multiple interfaces. The following description is the result of multiple *Design-Implement-Analysis cycles* . Each of the interfaces was processed and implemented separately before they were brought to accordance. Since the components do not affect each other, this kind of development ensures a simplified troubleshooting. Among others, our goal was to create a preferably compact *Multi-Modal-Tangible*, which can be easily reproduced and expanded.

3.1 Design and Fabrication

This section will concentrate on the Multi-Modal-Tangibles hardware (working title: *molT*). It will be discussed why and how the individual components are used. While doing so, we will name emerging advantages of a tangible with these components.
The chosen order simulates the actual way to build molT.

Ground

We started by using a classic PERCs ground. It consist of a 4mm thick plexi glass which has several holes. One use of

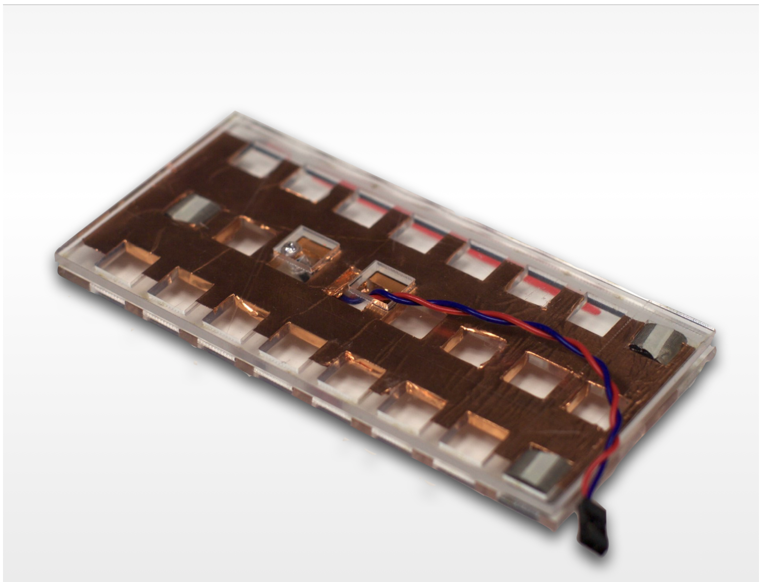

Figure 3.1: PERCs ground

PERCs ground to get detected by the touchscreen

the holes is to place the touchmarker in it. These are connected with copper and ensure the tangible's detection by the touchscreen. The area covered by copper has also a second role: It is used as a field sensor to notify the system, as soon as the tangible is placed on a capacitive touchscreen. There is a correlation between the copper's area and the distance in which the tangible gets detected as 'on screen'. We decided to use a larger area and adjusted the sensitivity on the software side. By using different thresholds, developer can now compute the approximate vertical distance of the tangible to the touchscreen. This might be useful for some use cases.

The other use of the holes is to stick the light sensor through it. This sensor is necessary to get detected reliably by the *multi-touch surface* in the correct orientation. Figure 3.1 shows all of the mentioned components combined. Since the following components of molT are placed on top of this ground, it specifies the final size of the tangible, which is 140mm wide and 65mm long. This is, disregarding the height, the size of an iPhone 7.

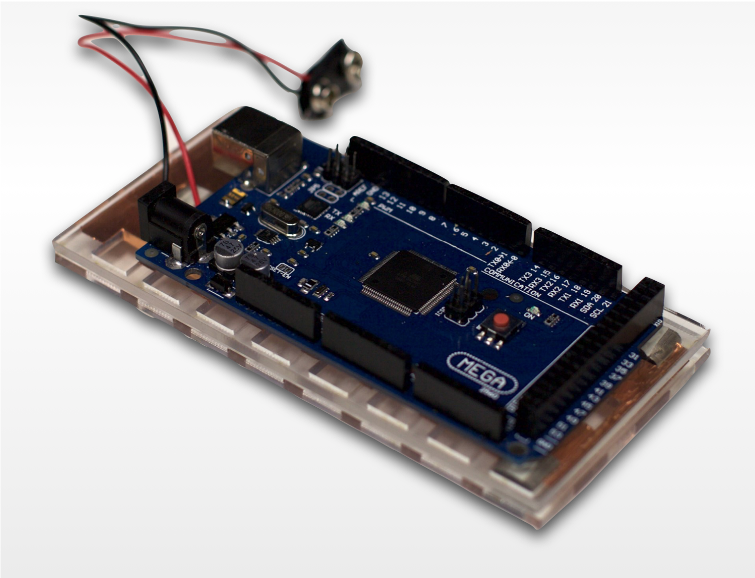

Figure 3.2: Arduino Mega

Arduino Mega 2560

As visible in Figure 3.2, the size of the ground is chosen so that an *Arduino Mega 2560* can be placed on top of it. We chose an Arduino because it has two major advantages.

1. It is **'ready to use'**.
   Arduino comes in a complete package form which includes the 5V regulator, an oscillator, a micro-controller, serial communication interface, LED and headers for the connections. Furthermore, developer do not have to think about programmer connections.

2. A **large community**.
   Many examples of code and libraries are accessible which are either provided or approved by Arduino.

Arduino Mega has suitable characteristics & provides easy deployment

The reason we use the Mega is because it has a high number of pins. Since moIT's display already takes 28 pins (including *Vcc* and *GND*), we needed a board with more than 30 pins to cover the other components. This leads us to the

Mega with 51 pins and multiple Vcc and GND pins. Another advantage of the Mega is its 256kb memory. The final sketch size of molT is over 34kb, so the *Arduino Uno* with its 32kb and many others are out of question.

Nevertheless, the Mega has disadvantages as well. On the one hand its physical size, which is almost unavoidable because of the pins, and on the other hand the fact that its built-in *comparator* is not connected to any of its pins (AtmelCorporation [2014]). Since the PERCs setup requires a comparator, an own one has to be added to the system. The comparator is used to compare the (by the field sensor) given voltage with a reference voltage, to determine whether the tangible is placed on a capacitive touchscreen or not. As different comparators have a different level of sensitivity, the circuit had to be adjusted.

Built-in comparator is not usable

Thus, we needed to work out a way to manipulate the sensitivity of the comparator. By increasing the difference between the reference voltage and the measured voltage, it is possible to decrease the comparators sensitivity. Figure 3.4 shows a resistor (R1) we build in to produce this causality. By increasing R1, the sensitivity decreases and vice versa.

Bluetooth LE

For the communication between tangible and touchscreen we added a *Bluetooth Low Energy (BLE) module*. The key benefits are its small size as well as its low power consumption, which are desired properties for a tangible. Just like Bluetooth, BLE operates in the 2.4 GHz ISM band. Contrary to classic Bluetooth, BLE remains in sleep mode constantly except for when a connection is initiated. The actual connection times are only a few ms, unlike Bluetooth, which would take $\sim$100ms. The reason the connections are so short, is that the data rates are as high as 1 Mb/s. Since we installed components which are triggered via Bluetooth, high data rates were necessary to provide a preferably low latency. Hence, why we used a module called AC-BT-V4, visible in Figure 3.3, and connected it to one of the three possible serial ports of the Mega (Serial3).

BLE provides low power consumption and high data rates

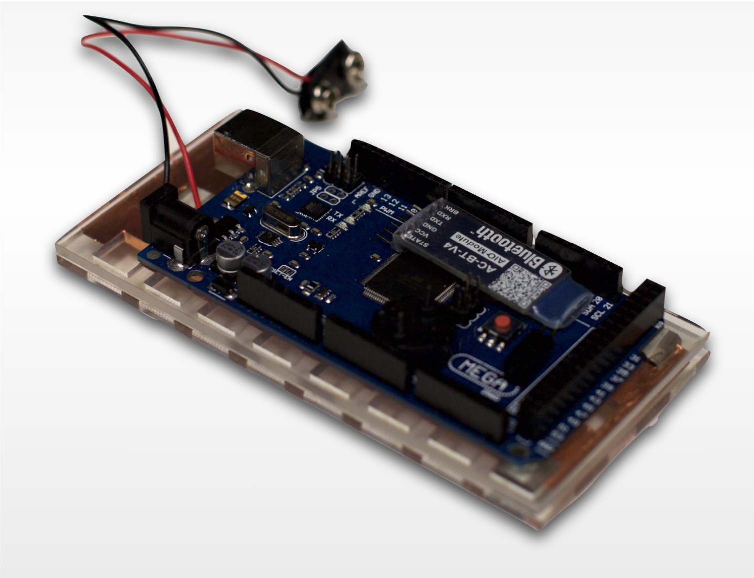

Figure 3.3: Positioning of BLE, speaker and tilt sensor

Speaker & Tilt Sensor

Speaker to strengthen the representation

Although tangibles represent real world objects, their representation is not perfect. For example, while playing Air hockey in real world, the mallet and puck permanently collide with each other. This collision creates a sound, which has its source right where the collision happens. The tangible-mallet (visible in Figure 1.1) will not be able to create such sound, because the collision only exists in the system. Of course, the computer could simulate the sound, but this would not be as realistic as a sound having its source right where the collision happened. Thus, to strengthen the representation of the tangible, we installed a speaker. Now the tangible not only looks like its representation, but sounds like it as well.

Able to detect motion gestures

In other use, it can be beneficial to detect *motion gestures* on a tangible. For example, if the tangible represents a glass of water, it is now possible to tilt it to drain its content. For our purposes, we need to detect shaking, so that a tilt sensor, which is a simplified version of an acceleration sensor, is sufficient.

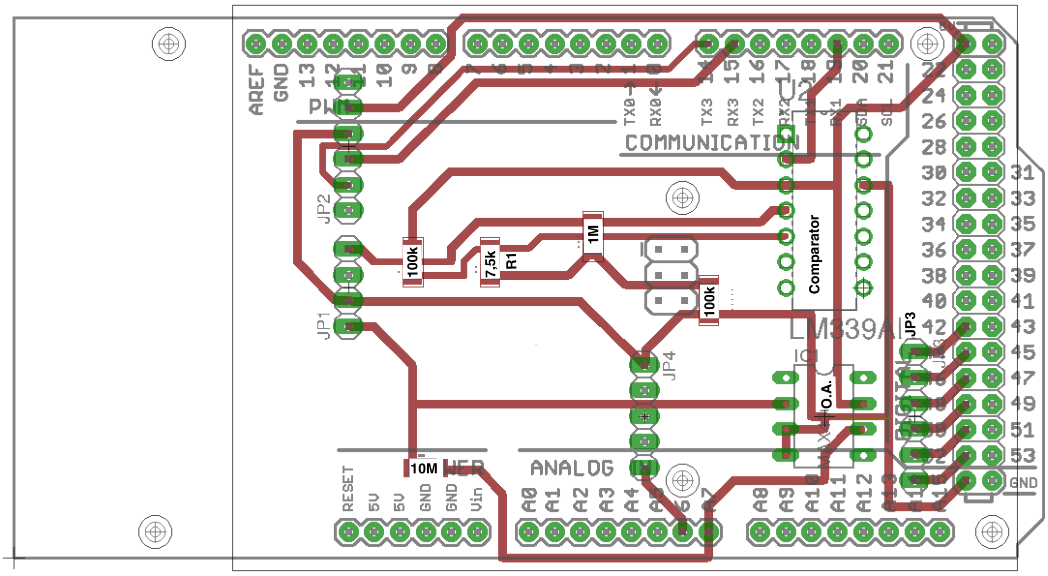

Figure 3.4: Shield design

In order to save space BLE, speaker and tilt sensor are placed directly on the Arduino's ground (presented in figure 3.3) and connected to the following shield.

Shield

Connecting the individual components

The shield's main function is to connect the installed components. Also integrated is an *operational amplifier*, which is necessary for the light sensor (Voelker et al. [2015]), and the already mentioned comparator. Figure 3.4 shows the allocation of the components related to the underlaying Ardunio Mega. A 'JPx' pin header is the interface to the individual components. JP1 is used by the light- and field sensor, JP2 by the BLE, JP3 by tilt sensor and JP4 by the speaker. The shield is created, adjusted and iterated with a tool called *Eagle*. Eagle has, among other features, a schematic capture editor and a printed *circuit board* (PCB) layout editor. This allows us to draw the circuit first and place the items as wished afterwards. To save place, we try to avoid components right under the display. Instead, we placed the comparator and operational amplifier right next to the screen, as shown in Figure 3.5.

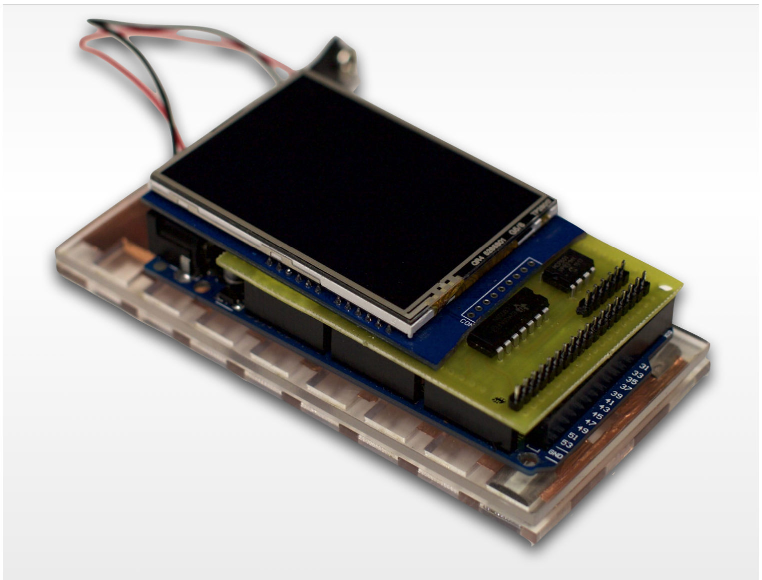

Figure 3.5: Touchscreen on top of the shield

Touchscreen

The presented display is a resistive TFT 2.8" touchscreen. It has a built-in *microSD* slot which is theoretically accessible through the pins. Unfortunately, every producer uses different controller for their display shield. Furthermore, different Arduino's expect certain pins to be placed correctly on the board. Without more ado, it is not possible to just place the display on the board and to use it. The necessary software adjustments will be discussed in section 3.2.

Illustrate the current representation of the tangible

The display can be used as an input method by implementing a *gesture recognition*. Although the main use is to show the current representation of the tangible by loading images (with BMP format) from the microSD card. The display is an essential component, since it is the only way to literally display the current representation of the tangible. The importance of having a preferably obvious representation goes hand in hand with the most important characteristic of tangibles, which is providing a suitable physical representation for digital information.

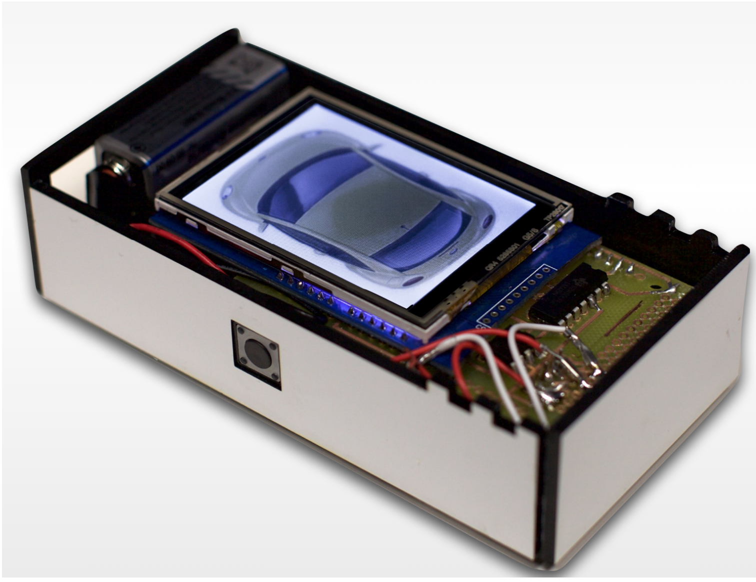

Figure 3.6: Case design

Vibration & Button

Vibration extends haptic feedback for tangibles

Last but not least, adding vibration and button to the case makes molT almost complete. Due to their shape, tangibles already allow *eyes-free interaction* with multi-touch surfaces. Therefore, it may occur that the user's focus does not lay on the tangible during the interaction. In this situations, vibration as an extension of the already existing haptic feedback, provides the possibility to transmit more information to the user.
The vibration motor is placed on the inner side of the case. Figure 3.6 shows two white cables which tail to the vibration motor. This enables a strong feedback close to the users hand.

Button as haptic input method

The other component assisting the eye-free interaction is the button. Currently, our button is positioned so that a right handed user can easily push it with his thumb. In case someone prefers the left hand, molT is built symmetrically so that we can rotate the display to place the button on the right.

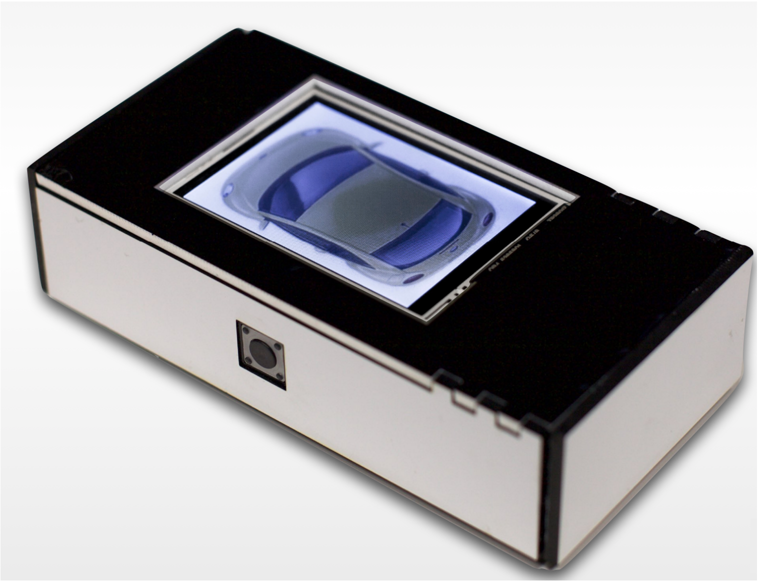

Figure 3.7: Final view of molT

Furthermore, Figure 3.6 illustrates the positioning of the energy source. We used a standard 9V battery which provides molT with energy for at least 2 hours. For the programmer connection to the Arduino we left a hole in the case (hinted in Figure 3.6). In case of using a rechargeable battery instead of the battery, this hole can be used to recharge the tangible.

Case

Since the case forms the representation of the tangible, we kept the design plain and neutral to place the focus on the display. As previously explained, we designed the case symmetrically in order to provide a comfortable use for left- and right-handers, but also because of esthetically reasons.

Plain and neutral case design to place focus on display

The case, including the ground, is designed with a tool called *OpenSCAD*. This script based modeler was used to create *SVGs* which were printed by a *laser cutter*. Afterwards, everything but the top is glued together. We left a removable top to enable access to all of the components without building a new case.

3.2 Software

Define the used pins

In general, the first step of programming an Arduino is to define the used pins. We were free to choose the pins for every of our components but the touchscreen. Because of its shape, the touchscreen can only be connected to the Arduino in a certain way. Due to this premise and the fact that some of the display pins have to be connected to special Arduino pins, it is not possible to use the touchscreen without further adjustments.

Every Arduino has a *Serial Peripheral Interface (SPI) bus,* which is depending on the board connected to different pins. This bus is used for short distance communication, primarily in embedded systems. In our case, it is used to establish the communication to the touchscreen and its microSD card slot.

Emulate the SPI pins for the Arduino Mega

While the SPI Bus of the Arduino Mega is connected to the pins 50-52 (or alternatively 1,3,4) [SPI], our touchscreen uses the pins 11-13, thus we can not use the hardware implementation and are forced to emulate it with software. The emulation was implemented by [Greiman] in 2009. Although it enables the use of arbitrary pins for the SPI bus, further adjustments are necessary for the Arduino Mega. The usage of the hardware pins must be explicitly commented out in the SD Library file *SD2Card.h*. This causes a compile error because the function *setClockDivider()* is not defined anymore. Since it is also not used anymore this function can and must be deleted as well. Now we can map the SPI bus to the needed pins 11-13.

Use the internal pull-up resistor

Continuing defining the pins, we have two options for the button and tilt sensor: Either using the classic circuit which is three pins (Vcc, GND and arbitrary pin) and an external pull-up resistor [Ard], or using two pins (Vcc and arbitrary pin) and an internal pull-up resistor. The latter are resistors that connect to Vcc internally, provided by the Atmega

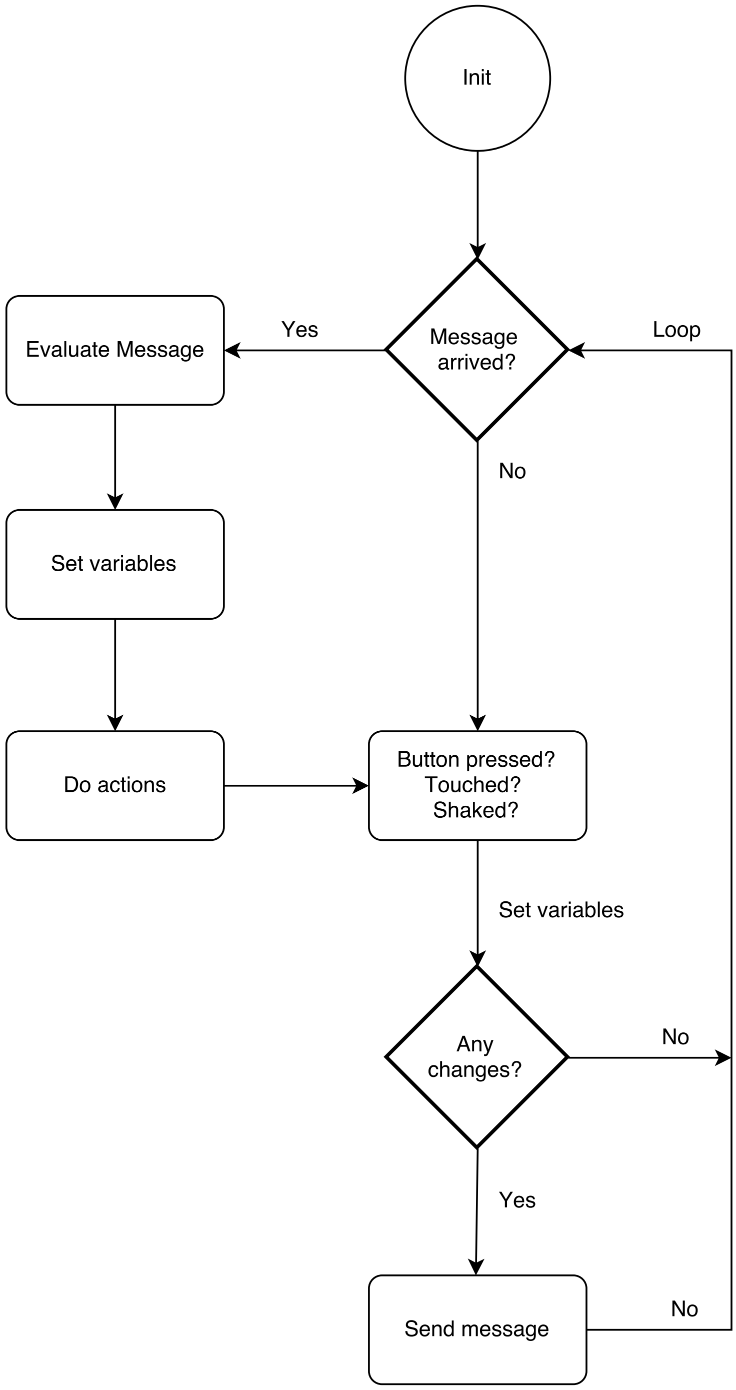

Figure 3.8: Flowchart of arduinoprogram

microcontroller. Since we desire to save pins and reduce the circuit we decided for the latter solution.

Once every component is initialized the program enters a loop which is illustrated in Figure 3.8. The loop is divided into three main functions:

1. Look for arriving messages.
   At the beginning, we examine whether there is new processable information arriving. If that is the case, we store the content in the corresponding variables and execute the belonging actions.

2. Detect user input.
   We also track permanently if the user has entered any input. Each of the states are stored in variables which can be sent if necessary.

3. Send out changed properties.
   To reduce the traffic and thus the power consumption, we only send messages if the properties have changed. Since Bluetooth provides a reliable connection (Andersson [2013]), we can use this form of communication without losing any information.

Chapter 4

The Game

Figure 4.1: Using moIT

To see and feel the performance of a Multi-Modal-Tangible it takes a use case scenario which is particularly made for it. Our focus lies on creating an intuitive connection between tangible and use case. Thus, we decided to create a variation of the well-known game Space Invaders. However, our game is modified in that way, that a controller with input and output is required to manage the game. Figure 4.1 shows a snapshot of someone playing with moIT.

Creating suitable use case for moIT

moIT represents a vehicle, which has to be changed as soon as the game environment changes (illustrated in Figure 4.2). While playing, the user will try to avoid bullets and also has the opportunity to use a shield to defend himself. If the player always uses the correct vehicle for the current environment and has not run out of lives after 120 seconds, he wins the game.

Variation of Space Invaders

4.1 Setup

The game is played on a Microsoft multi-touch-table. To detect the touchpoints of the tangibles a framework called MultiTouchKit by Linden [2015] is used. It is written in Objective-C and uses Cocoa, which is Apple's native object-oriented application programming interface (API) for their operating system OS X.

Using MultiTouchKit and SpriteKit to develop the game

The game is developed in the same environment using a framework offered by Apple called SpriteKit. SpriteKit provides a graphics rendering and animation infrastructure that can be used to animate arbitrary textured images or sprites. Besides for background and labels, we used this framework to create sprites, in form of bullets, and a shadow sprite positioned right under the tangible.

Enable physics for bullets and player

There exist two types of bullets in our game. Larger bullets, which are spawn randomly and smaller bullets, which form an unavoidable line, both moving towards the player. Both types of the bullets are visible on Figure 4.2. The invisible shadow of the tangible is used to have an object in the size of the tangible, which represents the tangible to the system.

Now, the provided physics engine can be used to detect collisions between bullets and shadow sprite. By turning on physics and setting a collision bitmask, it can be defined what happens if the bullet's and players physics body touched. In our case, the players life count decreases.

Give behavior to the elements

Another action, that happens in certain intervals, is the change of the environment. Announced with a countdown (visible on the left side of figure 4.2), the background will change and thus the player is in need for a different vehicle. At this point, the system waits for an arriving message from the tangible, telling that the correct vehicle is chosen. Otherwise, the player will lose the game. To create a communication between tangible and table, the MultiTouchKit had to be extended. Beforehand, the communication only worked in one direction. The framework expects the tangible to send on one certain characteristic whether the tan-

Figure 4.2: Two snapshots of the game

gible is placed on the table or not, and whether the light sensor has detected white light or not. Since we need to send and receive more information, a new Bluetooth communication had to be set.

Need of an extended communication

Bluetooth Communication

The connection is set on demand. First, we start up a central manager object, which is responsible for setting up the connection. The central manager searches for peripheral devices that are advertising. In the form of advertising packets, peripherals broadcast some of the data they have. An advertising packet is a relatively small bundle of data. It contains useful information about what a peripheral has to offer, such as the peripheral's name and primary functionality. If one of the found names matches with the requested name, CoreBluetooth [Cor] and IOBluetooth [IOB] (Apple frameworks) are used to set up the connection.

Creating connection by using CoreBluetooth and IOBluetooh

<table>
<tr><td>Tangible</td><td>→</td><td>Table</td></tr>
<tr><td colspan="3">”onTable#lightSensor#buttonPressed#state#shaked</td></tr>
<tr><td>Tangible</td><td>←</td><td>Table</td></tr>
<tr><td colspan="3">”vibrate#playSound#requierdState”</td></tr>
</table>

Table 4.1: Communication Protocol

Once connected, we explore the data on a peripheral device and send, read and write requests to a characteristic value of the peripheral’s service.

Finally, by subscribing to a characteristic’s value, we get notified when a message arrives. The arriving message will be evaluated and stored globally. If necessary, e.g if the player collides, a *sender* object is used to send a vibration request to the tangible. Table 4.1 shows the communication protocol, which is now easily expandable.

On each side, the string is split by ”#”, and the values are stored in the corresponding variable. This ensures, that the current values are accessible at any time.

4.2 Gameplay

Mapping tangible components and the game

Once the communication is set, the user can start playing. To avoid the bullets, the tangible has to be moved. If he, however collides, the tangible will **vibrate**. To protect against the line of bullets the player has to press the **button**. However, the player does not have unlimited protection. Once he is defenseless, a warning signal will **sound**, which should be a reminder to refill the protection by **shaking** the tangible. Finally, the **touchscreen** is used to change and simultaneously display the representation of the tangible. This is an approach to use the tangibles input and output interfaces appropriately. To see whether user discover these functionalities intuitively or not and how appropriate this allocation is, we have conducted a user study.

Chapter 5

Evaluation

Tangibles have the advantage that their functionalities are intuitively perceptible. This follows from the fact that their only functionality is bounded to their representation. This leads to the following question: Will the perception of the functionalities remain intuitively, if we expand the functionalities of tangibles?

No explanation needed for moIT?

The goal of this study is to find out what influence Multi-Modal-Tangibles have on the interaction with capacitive touchscreens. To discover whether there is a correlation between the complexity of tangibles and an intuitive perception, we have asked 20 participants to play a variation of the popular game Space Invaders, presented in chapter 4.

5.1 Model Extraction

Participants try to discover moIT's features

Since using tangibles usually does not need further explanation, the experimental setup does not foresee explanation either. Participants played without any instruction and tried to win the game. An appropriate evaluation method for this setup is Model Extraction. At this, user were supposed to explain the elements while using them and afterwards evaluate them as well. Playing this game without using all functionalities of molT, makes it impossible to win. The game is designed in a way that during the game, the player gets into situations in which he feels the need for certain components. Through that we evaluate whether the player falls into the *gulf of execution* (Norman [2002]) or not. This term describes the gap between a user's goal for action and the means to execute that goal. After finishing the game, the participants were asked to take a stand on this statement:

Searched for gulf of execution

It was easy to discover what molT can do.

- totally disagree: 0
- disagree: 3
- agree: 14
- totally agree: 3

The results show that although more than the majority of the participants agree with the statement in principle, something stop 17 of 20 participants from agreeing totally. To check whether it is because the game created misleading affordances or because it was unclear how to operate the tangible, we asked them to take stand to the statement:

Searched for gulf of evaluation

The different ways to interact with the tangible were fitting the gameplay.

- totally disagree: 0
- disagree: 1
- agree: 8
- totally agree: 11

Evaluating individual components

Therefore, it can be concluded that the gulfs of execution do not lead back to the gameplay. Thus, it was also part of the questionnaire to evaluate the individual components and determine which of them are useful for tangibles. Hence, the participants were asked, based on their intuition, whether they think that the individual components are useful input and output methods for tangibles or not.

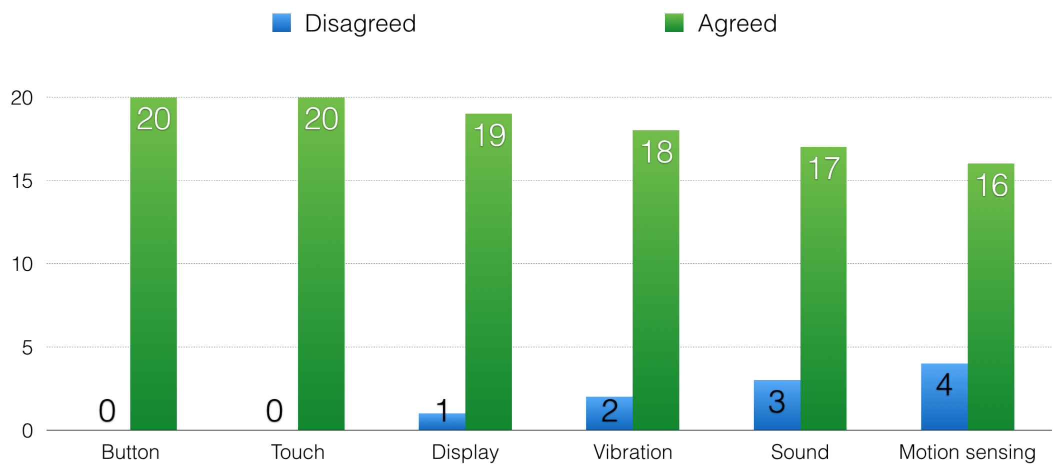

Figure 5.1: Statistic results of study

The answers are illustrated Figure 5.1 showing that, almost everybody agreed to the button touchscreen and display feature. But in case of basically invisible components, there were some participants who disagreed. For example, they said motion sensing it not intuitively discoverable, mainly because it is not visible and they don't know that tangibles are able to do so. Participants who disagreed regarding sound had the opinion that during the gameplay, they would notice where the sound cames from, so it would not be necessary to build in an extra speaker for the tangible.

moITs functionalities are intuitively discoverable

Overall, the results show that using moIT does not need further explanation, which is a really desirable characteristic for a tangible. But developer must consider that the design process for Multi-Modal-Tangibles does not end by design the tangible itself. It is in addition important to create the correct and matching affordances in the particular use case.

5.2 Controlled Experiment

Considering the assumption that haptic feedback on a tangible is (like the previously mentioned sound) for some

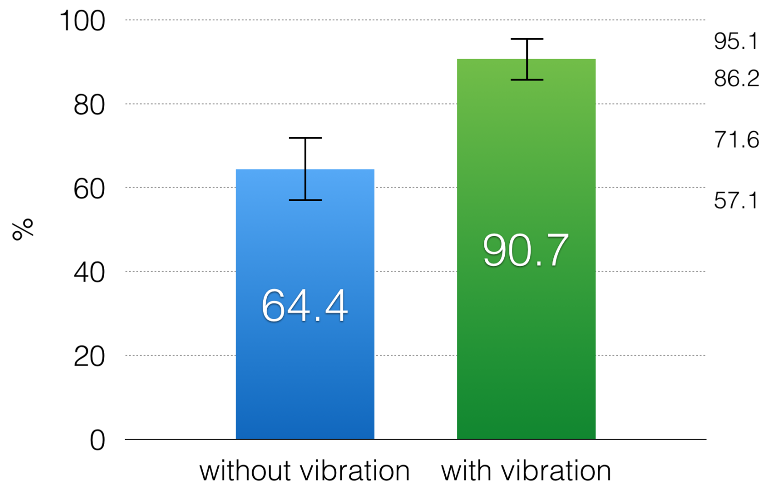

Figure 5.2: Accurancy with & witout haptic feedback

people an unnecessary component, we added a second part to our study. We evaluated whether haptic feedback provides an enhanced interaction for tangibles or not. The results should help to discover if that kind of feedback enhances the interaction significantly.

Therefore, we made a controlled experiment and let the participants play one with and one without vibration. Each time they were supposed to count their collisions. Figure 5.2 shows how close their counted number comes to the actual number of collisions. Also visible is the 90 % confidence interval for each of the trials.

haptic feedback for tangibles provide significant better perception

The results show that even the lowest expected accuracy of playing with vibration (86.2 %) is still 14.6 % higher than the highest expected accuracy of playing without vibration (71.6 %). Since in addition participants were in average about 26 % more accurate, it can be said that vibration for a tangible enables a significant better perception.

Chapter 6

Summary and future work

The function of tangibles starts and ends with their physical appearance. Regarding the fact, that tangible interaction is becoming increasingly important for Human-Computer Interaction, we felt the need of extending the possible ways of interaction.

6.1 Summary and contributions

In this thesis, we introduced a Multi-Modal-Tangible based on the concept of PERCs. Our prototype includes speaker, vibration, display as output and touchscreen, motion sensing and a microphone as input. Based on this prototype we have created a game to evaluate the interaction with a Multi-Modal-Tangible. Our goal was to discover whether a complex tangible is still intuitively usable like single-modal tangibles or not. Participants evaluated the individual components and showed a greater acceptance regarding visually recognizable components.

Idea, development and evaluation for a Multi-Modal-Tangible

Our approach shows that designing tangible user interfaces includes not only designing the tangible and the use case. It is also important to bring these two components into accordance and create the right affordances to enable an intuitive understanding of the system. Our evaluation showed that if the installed components are used so that they assist the representation, user also discover basically invisible features.

6.2 Future work

Integrate into more use cases

We have several ideas, how researchers could build upon our work for future research. First of all by finding and creating use cases where molT could fit. Thereby it could happen that either not every or more interfaces are needed. At this, it is handy that our approach can be easily adjusted or extended.

Provide more interfaces

We asked our participants whether they could think of any extension and some of them suggested a pressure sensor. Their idea was based on the iPhones 3D Touch [3DT] function and they supposed it for molTs touchscreen. Continuing this idea, we think of a pressure sensor that could be added to the tangible. This would enable an additional way to interact with the screen.

Enhancing the prototype

Thus the prototype itself can be enhanced as well. To enforce the tangibles representation, molT could be combined with the display blocks concept presented by Pla and Maes [2013]. This means that every side of the tangible consist of displays. Such a tangible makes it possible to display a 3D representation of the tangible. For this and in general a higher performing processor could be used.

Appendix A

Appendix for the model extraction and the controlled experiment

This appendix includes:

- The consent form that the participants had to sign prior to the study (A.1)
- The questionnaires we handed out after our experiment (A.2 & A.3)

Informed Consent Form
moIT: Multi-Modal Tangibles to Enhance Interaction with Capacitive Touchscreens

PRINCIPAL INVESTIGATOR:Armin Mokhtarian
Media Computing Group
RWTH Aachen University
Phone: +4917645652816
Email: armin.mokhtarian@rwth-aachen.de

Purpose of the study: The goal of this study is to find out what influence Multi-Modal-Tangibles have on the interaction with capacitive Touchscreens. Participants will be asked to play a variation of a popular game Space Invaders. We will measure if any of the functionalities provide an enhanced interaction experience.

Procedure: The study will take approximately 10 minutes. Participants will play a game on a Multi-Touch-Table, using a Multi-Modal-Tangible. Participants should try to beat the game. In some trials they will count count how often they collide. There will five trial rounds.

Risks/Discomfort: You may become fatigued during the course of your participation in the study. You will be given several opportunities to rest, and additional breaks are also possible. There are no other risks associated with participation in the study. Should completion of either phase prove discomforting, it will be terminated immediately.

Benefits: The results of this study will be useful to discover if the use of a Multi-Modal-Tangible is intuitive. It will also help to determine if haptic feedback on a Tangible provides a better interaction experience.

Alternatives to Participation: Participation in this study is voluntary. You are free to withdraw or discontinue the participation.

Cost and Compensation: Participation in this study will involve no cost to you. You will be provided with drinks and chocolates to get you through the study ☺

Confidentiality: All information collected during the study period will be kept strictly confidential. You will be identified through identification numbers. No publications or reports from this project will include identifying information on any participant. If you agree to join this study, please sign your name below.

_____ I have read and understood the information on this form.
_____ I have had the information on this form explained to me.

Participant's Name Participant's Signature Date

Principal Investigator Date

If you have any questions regarding this study, please contact Armin Mokhtarian at +4917645652816
email: armin.mokhtarian@rwth-aachen.de

Figure A.1: Consent form informing the participant about the purpose and structure of the study.

User Study

moIT: Multi-Modal Tangibles to Enhance Interaction with Capacitive Touchscreens

Armin Mokhtarian

Age: __________

Gender:
☐ M ☐ F

1. Using moIT was a new way for me to interact with Touchscreens.

☐ Totally disagree ☐ Disagree ☐ Agree ☐ Totally agree

2. It was easy to discover what moIT can do.

☐ Totally disagree ☐ Disagree ☐ Agree ☐ Totally agree

3. The different ways to interact with the tangible were fitting gameplay.

☐ Totally disagree ☐ Disagree ☐ Agree ☐ Totally agree

3a. The **Display** helped me to visualize the current state of the controller.

☐ Totally disagree ☐ Disagree ☐ Agree ☐ Totally agree

3b. **Buttons** are useful input methods for tangibles.

☐ Totally disagree ☐ Disagree ☐ Agree ☐ Totally agree

3c. **Touchscreens** are useful input methods for tangibles.

☐ Totally disagree ☐ Disagree ☐ Agree ☐ Totally agree

1

Figure A.2: First page of questionnaire.

3d. **Movement sensing** is a useful input method for tangibles.

☐ Totally disagree ☐ Disagree ☐ Agree ☐ Totally agree

3e. **Displays** are useful output methods for tangibles.

☐ Totally disagree ☐ Disagree ☐ Agree ☐ Totally agree

3f. **Sound** is a useful output method for tangibles.

☐ Totally disagree ☐ Disagree ☐ Agree ☐ Totally agree

3g. **Vibration** is a useful output method for tangibles.

☐ Totally disagree ☐ Disagree ☐ Agree ☐ Totally agree

4. I had fun playing with moIT.

☐ Totally disagree ☐ Disagree ☐ Agree ☐ Totally agree

5. What did you calculate as your Collision Count for each scenario?

First scenario: _____ ***Second scenario:*** _____

6. Can you think of any other input/output modalities that could be added to tangibles:

7. Comments:

Thank you very much for taking the time to complete this survey. Your feedback is valued and very much appreciated!

2

Figure A.3: Second page of questionnaire.

Bibliography

Apple iphone 3d touch. https://help.apple.com/iphone/10/#/iphcc8f419db. Accessed: 2016-09-24.

Arduino button. https://www.arduino.cc/en/Tutorial/Button. Accessed: 2016-09-24.

Apple corebluetooth. https://developer.apple.com/library/content/documentation/NetworkingInternetWeb/Conceptual/CoreBluetooth_concepts/AboutCoreBluetooth/Introduction.html. Accessed: 2016-09-24.

Apple iobluetooth. https://developer.apple.com/reference/iobluetooth. Accessed: 2016-09-24.

Arduino spi library. https://www.arduino.cc/en/Reference/SPI. Accessed: 2016-09-24.

Mats Andersson. Using bluetooth technology in a demanding environment where reliability, robustness and interoperability are key success factors, 2013.

AtmelCorporation. Atmel atmega640/v-1280/v-1281/v-2560/v-2561/v datasheet, 2014.

Bakker and Hollemans. Weathergods: tangible interaction in a digital tabletop game, 2007.

William Greiman. Arduino sd library. https://www.arduino.cc/en/Reference/SD. Accessed: 2016-09-24.

Eva Hornecker and Jacob Buur. Getting a grip on tangible interaction: A framework on physical space and social interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 437–446, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7. doi: 10.1145/1124772.1124838. URL `http://doi.acm.org/10.1145/1124772.1124838`.

Hiroshi Ishii. The tangible user interface and its evolution. *Commun. ACM*, 51(6):32–36, June 2008. ISSN 0001-0782. doi: 10.1145/1349026.1349034. URL `http://doi.acm.org/10.1145/1349026.1349034`.

Hiroshi Ishii, Dávid Lakatos, Leonardo Bonanni, and Jean-Baptiste Labrune. Radical atoms: Beyond tangible bits, toward transformable materials. *interactions*, 19 (1):38–51, January 2012. ISSN 1072-5520. doi: 10.1145/2065327.2065337. URL `http://doi.acm.org/10.1145/2065327.2065337`.

Rong-Hao Liang, Liwei Chan, Hung-Yu Tseng, Han-Chih Kuo, Da-Yuan Huang, De-Nian Yang, and Bing-Yu Chen. Gaussbricks: Magnetic building blocks for constructive tangible interactions on portable displays. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 3153–3162, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557105. URL `http://doi.acm.org/10.1145/2556288.2557105`.

Rene Linden. Multitouchkit: A software framework for touch input and tangibles on tabletops and. Master's thesis, RWTH Aachen University, Aachen, September 2015.

Donald A. Norman. *The Design of Everyday Things*. Basic Books, Inc., New York, NY, USA, 2002. ISBN 9780465067107.

Pol Pla and Pattie Maes. Display blocks: A set of cubic displays for tangible, multi-perspective data exploration. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, TEI '13, pages 307–314, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1898-3. doi: 10.1145/2460625.2460677. URL `http://doi.acm.org/10.1145/2460625.2460677`.

Jun Rekimoto. Smartskin: An infrastructure for freehand manipulation on interactive surfaces. In *Pro-*

ceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '02, pages 113–120, New York, NY, USA, 2002. ACM. ISBN 1-58113-453-3. doi: 10.1145/503376.503397. URL `http://doi.acm.org/10.1145/503376.503397`.

Johannes Schöning, Jonathan Hook, Nima Motamedi, Patrick Olivier, Florian Echtler, Peter Brandl, Laurence Muller, Florian Daiber, Otmar Hilliges, Markus Loechtefeld, Tim Roth, Dominik Schmidt, and Ulrich von Zadow. Building interactive multi-touch surfaces. *Journal of Graphics, GPU, and Game Tools*, 14(3):35–55, 2009. doi: 10.1080/2151237X.2009.10129285. URL `http://dx.doi.org/10.1080/2151237X.2009.10129285`.

Simon Voelker, Kosuke Nakajima, Christian Thoresen, Yuichi Itoh, Kjell Ivar Øvergård, and Jan Borchers. Pucs: Detecting transparent, passive untouched capacitive widgets on unmodified multi-touch displays. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 101–104, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2271-3. doi: 10.1145/2512349.2512791. URL `http://doi.acm.org/10.1145/2512349.2512791`.

Simon Voelker, Christian Cherek, Jan Thar, Thorsten Karrer, Christian Thoresen, Kjell Ivar Øvergård, and Jan Borchers. Percs: Persistently trackable tangibles on capacitive multi-touch displays. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 351–356, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3779-3. doi: 10.1145/2807442.2807466. URL `http://doi.acm.org/10.1145/2807442.2807466`.

Neng-Hao Yu, Li-Wei Chan, Seng Yong Lau, Sung-Sheng Tsai, I-Chun Hsiao, Dian-Je Tsai, Fang-I Hsiao, Lung-Pan Cheng, Mike Chen, Polly Huang, and Yi-Ping Hung. Tuic: Enabling tangible interaction on capacitive multi-touch displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2995–3004, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979386. URL `http://doi.acm.org/10.1145/1978942.1979386`.

Index

Typeset October 4, 2016