

Using Tangibles to Teach Solutions to the Knapsack Problem

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

by
Marvin Bruna

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Ulrik Schroeder

Registration date: 12.04.2019
Submission date: 26.04.2019

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

Abstract	xiii
Überblick	xv
Acknowledgements	xvii
Conventions	xix
1 Introduction	1
2 Related Work	5
2.1 Framework and Tangibles	6
2.2 History of TUIs at the MIT	7
2.2.1 Early Stages of Tangibles at the MIT	7
2.2.2 From the Logo Turtle to LEGO Mindstorms	9
2.2.3 Music Blocks and Sensory Puzzles as true TUIs	10
2.3 Programming Robots with Tangibles	11

2.4	Children’s Fun Working with TUIs	12
2.5	A TUI for Teaching Neurosciences	13
2.6	Summary of Related Work	15
2.7	The 0/1 Knapsack Problem	16
3	Own work	19
3.1	First Design-Implement-Analyze Cycle	19
3.1.1	Flipbook and Paper Prototypes	20
3.1.2	Building Tangible Prototypes	21
3.1.3	Constructing the Different Problems	22
3.2	First Game Implementation	23
3.2.1	Graphical Design and Goal of the Game	23
3.2.2	Knapsack Solving and Input Implementation	25
3.3	Second DIA cycle	26
3.3.1	Building the final Tangibles	27
3.3.2	Choosing and Designing the final Problems	30
4	Evaluation	33
4.1	Intrinsic Motivation Inventory	33
4.2	Hypotheses	35
4.3	Design of the Study	36
4.3.1	Environment	36

4.3.2	Procedure	37
4.3.3	Measurements	39
4.4	Participants	40
4.5	Results	40
4.5.1	Quantitative	41
	IMI Questionnaire Rating	41
	Time to solve	42
	Additional design questions	43
4.5.2	Qualitative	44
4.6	Discussion	45
4.6.1	Pen-and-paper Approach	45
4.6.2	Tangible Approach	46
4.7	Design Recommendations	47
5	Summary and future work	49
5.1	Summary and contributions	49
5.2	Future work	51
A	User Study Consent Form and Questionnaire	53
B	Problems used in the User Study	61
	Bibliography	69
	Index	73

List of Figures

2.1	Seymour Papert	8
2.2	Logo robot turtle	8
2.3	Slot Machine by Perlman	9
2.4	Comparison of Music Blocks and blocks designed in " <i>Sensory Puzzles</i> "	11
2.5	Tern blocks	12
2.6	GUI and TUI puzzles	13
2.7	BrainExplorer setup and user interaction with the system	14
3.1	Flipbook prototype	20
3.2	Paper block tangible	21
3.3	Tangible with bird sand	22
3.4	Software after startup	24
3.5	User interface showcase and user interaction	27
3.6	Different tangible patterns	29
3.7	Improved PUC tangible	29

3.8	Tangible cover design	31
4.1	Intrinsic Motivation Inventory graph	42
4.2	Time to solve graph	43

List of Tables

2.1	Knapsack example problem.	17
2.2	Example table used for dynamic programming approach.	18
2.3	First step of the algorithm shown in the table.	18
2.4	Final table containing the correct solution. . .	18
3.1	Final weights of the tangibles in grams	30
4.1	Latin square used to determine the order of the conditions given to the user.	37
4.2	Overview of the 'Z' and 'p' values of the Wilcoxon signed-rank test.	41

Abstract

Tangibles have become increasingly popular in the learning and teaching environment over the last years. By providing additional haptic feedback, they can bridge the gap between virtual and physical object manipulation. Currently tangibles in the learning and teaching environment are mostly used for simple tasks and for computer science in particular, they are nearly exclusively used to teach basic programming operations. With this work, we will present a tangible approach to the abstract and complex part of computer science. We propose to use tangible not only for coding purposes, but also to teach and motivate students with regard to algorithmic problems. We do so, by looking at the 0/1 Knapsack problem, a famous abstract problem in computer science.

Therefore, a game is implemented and the currently existing tangibles are used as a basis to create new and improved ones, while we also tweak the underlying framework. We describe the design process of the software and the tangibles. Starting with early paper-style prototypes, going over a first software and hardware implementation and finally arriving at the finished project. While doing so, we will also discuss problems and their solutions working towards that finished version of the software.

Our conducted user study reveals significant improvement of the users intrinsic motivation. On average, users felt more enjoyment, more competence and less tension working with the tangible user interface. Additionally, the tangible approach also allowed users to solve the problems faster and helped them to figure out one heuristic to approximate the solution of Knapsack problems. Monitoring the tactics used and comments made by the participants, allowed us to derive further design recommendations for the problems and tangibles. By using tangibles to teach abstract computer science problems, the number of dropouts could be reduced, because tangibles seem to improve the intrinsic motivation, while also helping students to understand how to tackle abstract problems in computer science. Additionally, this thesis is only an entry point, as tangible could be used for a variety of abstract computer science problems.

Überblick

Tangibles haben in den letzten Jahren im Hinblick auf den Bereich Lernen und Lehren an Popularität gewonnen. Durch ihr haptisches Feedback schließen sie die Lücke zwischen virtueller und physischer Objektmanipulation. Momentan werden Tangibles im Lehrbereich meist nur für einfache Aufgaben verwendet. In der Informatik selbst beschränkt sich ihr Anwendungsgebiet nahezu ausschließlich auf das Beibringen von grundlegenden Programmierkenntnissen. Mit dieser Arbeit präsentieren wir einen tangible-basierten Ansatz für abstrakte und komplexe Probleme der Informatik. Wir schlagen vor, Tangibles nicht nur zum Programmieren, sondern auch zum Beibringen von algorithmischen Problemen und zum Motivieren von Studenten zu benutzen. Dies geschieht indem wir uns das bekannte, abstrakte 0/1 Rucksackproblem anschauen.

Dafür implementieren wir ein Spiel und bauen auf Basis bereits existierender Tangibles neue und verbesserte Tangibles. Außerdem justieren wir das zugrundeliegende Framework nach. Wir beschreiben den Designprozess der Software und der Tangibles. Der Designprozess startete mit Papier-Prototypen, ging dann über eine erste Software- und Hardwareimplementierung, um am Ende bei dem abgeschlossenen System anzukommen. Während wir diesen Prozess beschreiben, diskutieren wir außerdem Probleme sowie deren Lösungen, auf die wir im Verlaufe der Entwicklung zur finalen Version gestoßen sind.

Unsere durchgeführte Studie zeigt eine deutliche Verbesserung der intrinsischen Motivation. Im Durchschnitt fühlten Nutzer mehr Freude, höhere Kompetenz und niedrigere Anspannung wenn sie mit dem tangible Nutzerinterface gearbeitet haben. Zusätzlich erlaubte der tangible Ansatz den Nutzern die gestellten Probleme schneller zu lösen und half ihnen dabei eine Heuristik zum approximieren der Lösung von Rucksackproblemen zu finden. Die Beobachtung der Taktiken und Kommentare der Teilnehmer ermöglichte es uns außerdem, Designempfehlungen für die Probleme und die Tangibles zu präsentieren. Das Nutzen von Tangibles zum Lehren von abstrakten Informatikproblemen könnte helfen die hohe Anzahl an Studienabbrechern in diesem Fachbereich zu reduzieren, da Tangibles an-

scheinend dabei helfen die intrinsische Motivation zu verbessern. Zudem helfen sie Studierenden zu verstehen wie man an abstrakte Probleme der Informatik herangeht. Zusätzlich ist diese Studie als Einstiegspunkt zu betrachten, da Tangibles auch für viele weitere abstrakte Probleme der Informatik benutzt werden könnten.

Acknowledgements

First, I would like to thank all the participants that offered me their time and help in the user study. I would also like to thank Christian for all of his advice throughout the time spend on the thesis, giving me valuable lessons on where to improve and on what could be done differently. Last but not least, I want to thank my family and friends for their constant support, not only with regard to the thesis, but also in a more general manner for my whole study career this far.

Conventions

Throughout this thesis we use the following conventions.

Text conventions

Definitions of technical terms or short excursus are set off in coloured boxes.

EXCURSUS:

Excursus are detailed discussions of a particular point in a book, usually in an appendix, or digressions in a written text.

Definition:
Excursus

Variables used in the Knapsack explanation and some proper names are written in italic-style text.

myVariable or *myInvention*

Source code and implementation symbols are written in typewriter-style text.

`myClass`

In the thesis, "we" will be used instead of "I" or passive constructions at certain points. However, this does not mean, that this thesis was written by anyone other than myself and it only serves readability purposes.

The whole thesis is written in American English.

Chapter 1

Introduction

Computer science (CS) is part of the second largest field of study in Germany, engineering science ([Statistisches Bundesamt, 01/2019]). While the number of first semester engineering science students is high every year, so is the number of students quitting university over the course of their bachelor studies. For students, who would have finished their bachelor in 1999, 2002, 2004 or 2006 the dropout rate was: 37, 38, 39 and 32% respectively for the different years ([Heublein et al., 05/2008]). More recently this number went up to 46% for CS ([Heublein and Schmelzer, 2018]). Heublein et al. [2017] investigated the reasons, why students drop out of university in Baden-Wuerttemberg. Two of the main reasons revolved around performance and motivation problems, coming into university. The large drop out rates indicate that CS studies seem to be more difficult than other fields of study, and that students have even more issues to perform to their expectations and to motivate themselves to understand this complex field of study. Therefore, this study will look at an approach using tangibles, that can possibly help to motivate students and increase their understanding of abstract problems in CS.

A tangible user interface (TUI) allows people to interact with digital content through the usage of real-world physical objects, the so-called tangibles. Tangibles provide the user with a form of haptic feedback, that would otherwise be missing, working on a two-dimensional tabletop. Mod-

Currently many students quit computer science studies

How tangibles help connecting real-world objects with their digital counterpart

ern tangibles are recognized through a distinctive pattern. On multi-touch tabletops these patterns are build from conductive markers, and patterns for more traditional tabletops are build from fiducial ones in conjunction with a computer vision system that interprets them. Both ways allow them to act as an input method for the underlying virtual objects or other external objects, e.g., robots. They are also highly customizable, with regard to different use-cases, ranging from their shape, to weight and even their embedded hardware such as lights or error feedback through vibration. This allows for a broad use of tangibles when it comes to depicting different virtual objects, bridging the gap between virtual objects and real-world object manipulation.

Current use-cases
mostly for children

In recent history TUIs became more popular among researchers regarding their usage in a learning and teaching environment. Many of those research topics focus on children and therefore relatively simple tasks, from solving jigsaw puzzles to creating simple programs for educational robots (e.g. Xie et al. [2008] and Horn et al. [2012]). Both studies have shown positive results for the TUI as an input method for children and in the case of Horn et al. [2012] also for adults.

In addition to these use-cases, there were also studies that focused on more complex topics, including the use of tangibles to teach certain parts of neurosciences, where certain brain areas were depicted by tangibles and connected via virtual nerves on a tabletop ([Schneider et al., 2013]).

Lack of research that
focuses on tangibles
for algorithmic
problems in
computer science

The results of these studies support the idea, that TUIs are useful in a learning and teaching environment in general and also specifically in the field of CS. As mentioned earlier, Horn et al. [2012] already used tangibles to "code" simple programs for robots, but there is a lack of research, that focuses on the more theoretic aspect of CS, solving and working on algorithmic problems.

Outline

The goal of this thesis is to compare the use of a TUI against solving Knapsack problems manually with pen and paper, the method students have to use in assignments and exams as part of certain university subjects. We will start by

looking at the related work, where tangibles were used to learn or teach certain topics and the underlying framework we used. Next, we will take a short digression explaining the Knapsack problem itself, as well as the pen and paper method used to solve it manually. After that, we will discuss the design process of the software and the tangibles, that were used in the study, as well as problems that arose working with the underlying framework. The two different methods of working on Knapsack problems are then compared against each other in a user study, whose setup, procedure and results are presented. The results will be used to give further recommendations on the design of the tangibles, provide an entry point to the question, if a TUI is helpful when it comes to abstract CS problems and, if it can help to ease the mentioned issues students currently have, forcing a large number of them to drop out. A summary and suggestions for future work will finalize this thesis.

Chapter 2

Related Work

We will start by giving a short introduction to the framework used in the implementation, as well as a small look on the way current tangibles are build. Following this, we will summarize a roughly '30' year overview of TUI development from the Massachusetts Institute of Technology (MIT), that focused on educating children in general and in computer science topics, using TUI. Next, we will take a closer look at the three related work studies, we already mentioned in the introduction, as more recent and specific examples. First, we will look at how children performed using tangibles to program simple routines for an educational robot. After that, we will present the results of a study that focused on the self-reported enjoyment of children, solving jigsaw puzzles with different input methods. As the third and final example, we will look at how tangibles perform with regard to the more complex topic of neurosciences, to help students understand parts of this complex medical field. The history overview should give some insight on one development branch of tangibles in a learning and teaching environment, while the three explicit examples combined should serve as an entry point to this thesis, outlining the potential of tangibles to help solve a theoretic and abstract CS problem. We will wrap up the related work section with a brief recap on the Knapsack problem itself, creating the bridge to the main part of this thesis.

Overview

2.1 Framework and Tangibles

MultiTouchKit for
user input and
tangible
management

The framework used for the TUI to detect and process touch-based user inputs, is called *MultiTouchKit* (MTK) [Linden, 2015]. In addition, it can store and manage tangible properties, to determine their position on a multi-touch screen. The original version of the framework was written in Objective-C, but recently large portions were rewritten in Swift, to make it more accessible and future-proof. It is based on *SpriteKit* and offers two different input methods, mouse and network. A *Microsoft Surface Hub* acts as the tabletop, and the network source is used to send the touch inputs to the Mac. Other input methods, such as the tangibles, can be added in the *Tangible Configuration Center* (TCC), a software build from the framework. It is important to note, that every input source has to create instances of so-called `MTKTrace`. Traces represent continuous inputs and are, e.g., used as reference points for the tangibles. In the TCC, a set of exactly three such traces can be combined to create a marker pattern for a tangible. Adding a tangible pattern to the TCC and enabling it for your program (in the TCC as well) allows you to work with it in your own code and program.

Visual based
tangibles

Tangibles are currently built in multiple ways. Early constructions focused on the use of fiducial markers, black and white patterns, that get picked up and interpreted by a computer based vision system, e.g., the [reactIVision](http://reactivision.net)¹. The related work studies took this approach.

Passive untouched
tangibles for
capacitive
multi-touch screens

More recently tangibles are build to be used on capacitive multi-touch screens. The first way to build those tangibles is *Passive Untouched Capacitive Tangibles* (PUCs) [Voelker et al., 2013]. These follow a capacitive marker pattern approach, making them relatively easy to build, but also having the problem of being limited by these patterns, the underlying framework and the capacitive touchscreen itself. These were also problems we encountered and will discuss later on. The general construction consists of three conductive markers connected via copper foil. As long as the user touches this copper foil, the tangible is fully recognized, but

¹<http://reactivision.sourceforge.net/>

when the user stops touching it, it will eventually get filtered out by the in-build noise filters of the touchscreen.

The other approach focuses on *Persistently Trackable Tangibles on Capacitive Multi-Touch Displays* (PERCs) [Voelker et al., 2015]. Through the use of a sensor, the faint electric field emitted by every capacitive touchscreen can be detected, therefore it is possible for the tangible to know if it is still on the screen and just standing still, or if it actually has been removed. This information is sent via Bluetooth to the main system. Using the PERCs approach enables us to use a larger amount of tangibles, since we are not limited by the different patterns anymore and it also allows for even more customization.

Persistently trackable tangibles as another approach

2.2 History of TUIs at the MIT

McNerney [2004] provides a basic timeline of the development of TUIs at the MIT, that served learning and teaching purposes.

2.2.1 Early Stages of Tangibles at the MIT

Early approaches in the 1970s, saw Seymour Papert (see figure 2.1), the co-founder of the MIT Artificial Intelligence Laboratory, in cooperation with his students, develop a simple programming language for children, *Logo*².

Development of the Logo programming language

Logo is a variant of the List Processing (LISP) language family, that was also developed at the MIT in prior years. Logo allowed its users to create basic programs, to navigate a robotic turtle seen in figure 2.2.

After Logo had been taught for some years, it became apparent, that children had difficulties learning the concepts of programming, until reaching the age of 10-14 years. Radia Perlman then suggested two different input approaches

"Button Box" and "Slot Machine" as first steps to tangible approaches

²<http://el.media.mit.edu/logo-foundation/index.html>



Figure 2.1: Seymour Papert. Cropped version of the original photo taken by Rodrigo Mesquita on June 30, 2005. ([Mesquita, June 30, 2005]).



Figure 2.2: Two logo floor turtles. Original photo © 2004 by Robert W. Lawler ([McNerney, 2004]).

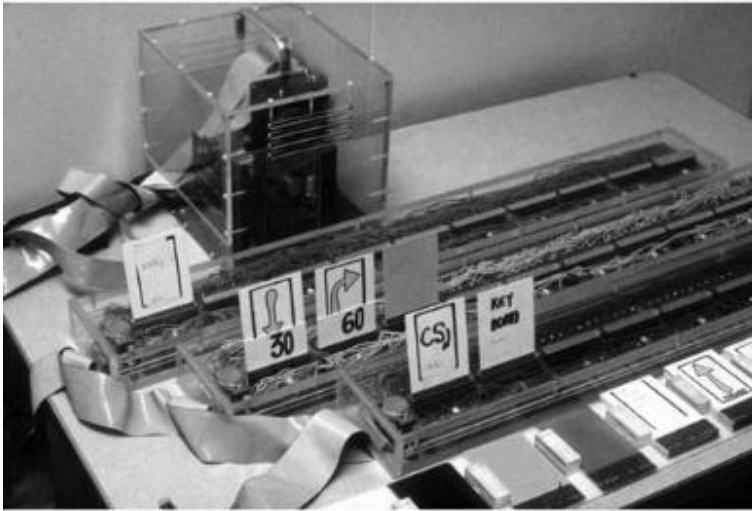


Figure 2.3: Slot Machine designed by Radia Perlman as an alternative input device for children. Original photo © 2004 by Robert W. Lawler ([McNerney, 2004]).

to allow younger children to program the turtle as well, informally being called the *"Button Box"* and the *"Slot Machine"* [Perlman, 1976]. The *"Button Box"* had several issues, e.g., if a mistake was made, the whole program sequence had to be redone. This was addressed in the *"Slot Machine"*, seen in figure 2.3, offering plastic cards for different programming operations. These cards can be viewed as the first steps towards TUIs.

2.2.2 From the Logo Turtle to LEGO Mindstorms

In the early 1980s, Logo gained popularity because personal computers (PCs) started to become available, in particular the Apple II. However, the robotic turtles were too "expensive, unreliable, and unsuitable" outside of pioneering laboratory work, so an on-screen turtle became the standard for geometry based Logo applications McNerney [2004]. Although the real-world turtle was not the main focus for some years, it made a comeback in the 1990s. Steve Ocko build the basis, for what would later become *Lego Mindstorms*. This allowed children to build Logo programs

First steps towards
Lego Mindstorms

on a PC, that would control devices build from LEGO blocks trough an "umbilical cord".

Removing the
"umbilical cord",
creating the original
Mindstorm blocks

Later, Fred Martin, Randy Sargent, and Brian Silverman "cut" this cord by creating LEGO blocks with an embedded computer, the birth of the first programmable LEGO bricks (Martin and Resnick [1993]). After LEGO acquired and commercialized the *P-Bricks*, they became known as a part of the *LEGO Mindstorms Robotics Invention System*. These robots are still used today, to teach children and even students simple programming techniques with or without tangibles.

2.2.3 Music Blocks and Sensory Puzzles as true TUIs

MITs' TUI, that was
build at the same
time as a commercial
toy with similar
functionality was
released

In 1999 Lackner, Dobson, Rodenstein and Weisman released their work on "*Sensory Puzzles*" [Lackner et al., 1999], a tangible approach for music based puzzles, albeit without a tabletop. At the same time, [Neurosmith](http://www.neurosmithtoys.com/)³ released a product, that was very similar to the *Sensory Puzzles* system. Both systems featured different blocks, containing small parts of Mozart's *Eine kleine Nachtmusik*, that could be arranged to either create the melody of the original or to create a new one. In addition, the blocks created by Lackner et al. [1999] also had their pitch, duration of notes and their tempo depicted by a "soundscape" on top of the block. A comparison of the blocks/systems can be seen in figure 2.4.

³<http://www.neurosmithtoys.com/>

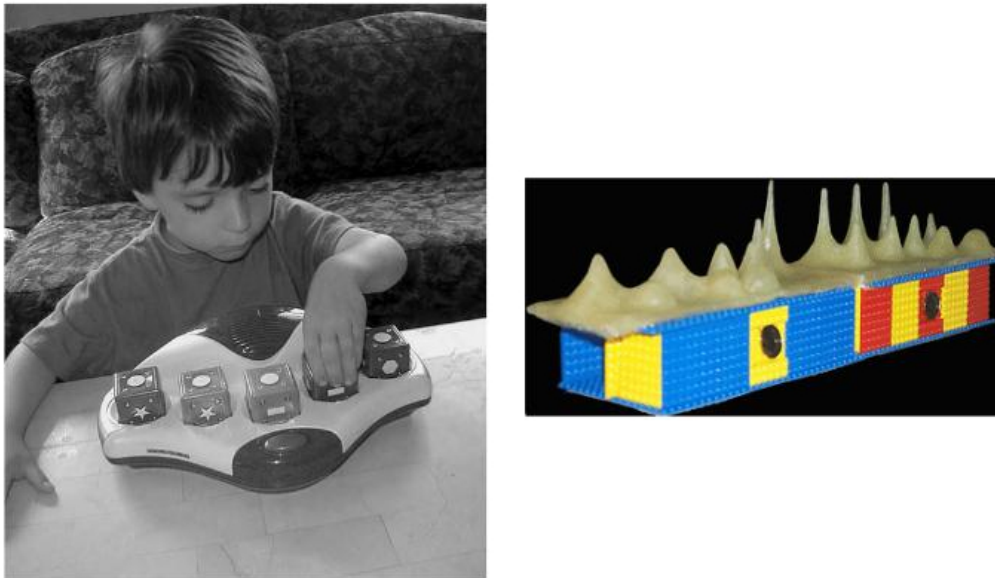


Figure 2.4: Comparison of the *Music Blocks* by Neurosmith (left) to the blocks designed by Lackner et al. [1999] (right).

2.3 Programming Robots with Tangibles

Today, many different robots (similar to the LEGO *Mindstorms* or the LOGO turtle) are used together with TUIs to teach children about programming. Horn et al. [2012] took a similar approach with their own system. They created *Tern* as their own tangible programming language and build different jigsaw-like blocks, that would depict different actions that the robot could perform (see 2.5). Users could stick the blocks together, creating a chain of program actions. Each block-type has a different marker, that could get picked up by a computer vision system, translating the blocks into actual commands for the robot.

Tangible blocks used to program a robot

Horn et al. conducted three different studies with mixed results. First, they tested their system at an exhibit at the Museum of Science in Boston. Next, they went to a kindergarten and finally, after further tuning their system, they tested it in a weeklong summer camp. Their first two studies showed, that tangibles proved useful when it came to enticing children to explore programming activities and for

Results of the study conducted by Horn et al.



Figure 2.5: Tern blocks used to create a chain of commands for a robot ([Horn and Jacob, 2007]).

whole-class activities and discussion, but they also noticed, that some of the best work was done in smaller groups with the Graphical User Interface (GUI). In the summer camp they discovered, that children could fluently switch between the two methods, preferring the TUI for early exploration and the GUI for “rapid prototyping” in later stages.

2.4 Children’s Fun Working with TUIs

Comparison of a tangible, graphical and physical user interface

Xie et al. [2008] conducted a study to investigate, if a TUI provided a more fun way of working on jigsaw puzzles. The study was designed for school-aged (seven to nine years old) children and compared a TUI against a GUI and a traditional Physical User Interface (PUI). Two different puzzles, one with a castle and one with a pirate ship, were rebuilt in the GUI and TUI. The GUI and TUI can be seen in 2.6. Again, the TUI pieces were recognized by a visual approach, using fiducial markers, that would get recognized by a computer vision software. Puzzle pieces did not have a full pattern themselves, instead connecting correct pieces



Figure 2.6: GUI on the left and TUI on the right ([Xie et al., 2008]).

together would make them recognizable. The TUI also provided similar audio and visual feedback, to what the GUI would display.

Xie et al. focused on collaboration between users and further used the Intrinsic Motivation Inventory (IMI) to assess the self-reported enjoyment, competence, choice and tension of the children. We will explain the IMI in one of the following chapters, since we also use it in our study. In their study no significant differences were found in the IMI scales, meaning that GUI, PUI and TUI nearly scored the same for all four mentioned scales over all three input methods. To be more precise, the TUI scored slightly higher in the enjoyment section and slightly lower on the three other ones. In addition, Xie et al. also reported that it took the children longer to solve the puzzles when using the GUI.

Results of Xie et al.
study

2.5 A TUI for Teaching Neurosciences

Over the last years, neuroscientists have learned more about the human brain and as a result topics such as "genomics, nanotechnology, advanced materials, cellular biology, and climate science" have become increasingly popular and will most likely find their way into more universities and maybe even schools. For that reason, Schneider et al. [2013] created a TUI, that was designed to help stu-

Usage of a TUI for a
complex topic

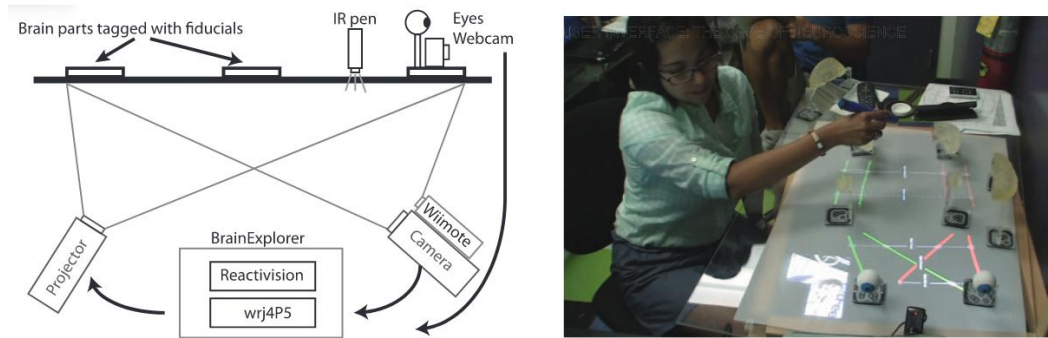


Figure 2.7: Left: Setup for the *BrainExplorer*. Right: User interacting with the TUI ([Schneider et al., 2013]).

dents understand parts of the large and still growing field of neurosciences, the so-called "*BrainExplorer*". Again, they used fiducial markers with a computer vision-based system to determine the position of the tangibles (polymer-based brain replicas). In addition, it is shown to the users what the brain would perceive and connections between different brain regions were displayed by a short throw projector underneath the table. An acrylic sheet with another sheet of velum on top provided a semitransparent surface and finally users were able to "*sever*" different connections by using an infrared pen, that would get detected by a Wiimote. The whole setup, as well as a picture of a user working with the TUI can be seen in 2.7 .

Results of the study

Schneider et al. used an AB/BA cross-over design, meaning a group A first worked with the TUI and a group B with traditional textbooks, then they took an interim test, after which they swapped the treatments and finally a post-test was conducted. They observed that the users who first worked with the TUI scored significantly higher, in both tests, than the group who first worked with the textbooks. Group A also established/uttered more rules than group B. These results suggest, that a TUI as an entry point to complex topics can help to build a knowledge base, that then could be further improved with actual textbook information.

2.6 Summary of Related Work

We started, by looking at one development branch that lead to TUIs in a learning and teaching environment. From the very early stages, with systems such as the *Button Box* or the *Slot Machine* [Perlman, 1976], over the construction of the first LEGO *Mindstorms*, that are still used in TUI studies or serve as a model for other programmable robots [Martin and Resnick, 1993], arriving at the first systems, that would be called a TUI, the *Music Blocks* and *Sensory Puzzles* [Lackner et al., 1999].

Summary of the MIT timeline

After showing this one timeline, we focused on three more recent and specific examples. First, we looked at a study by Horn et al. [2012], who build their own robot and tangible programming language, known as *Tern*. Consisting of puzzle-like blocks, with different fiducial markers, that were translated into actual program code by a camera and a corresponding software. They showed that a TUI provides a good approach when it comes to teaching children simple programming tasks. Programming is also one of the big topics in CS classes at universities, that is also linked to this thesis, as we will focus on the usage of a TUI in another large area of CS, abstract and algorithmic problems. Next, we looked at a study by Xie et al. [2008], who investigated children's self-reported enjoyment working with different input methods (GUI, TUI, PUI) to solve jigsaw puzzles. Their results showed no significant differences in the reported enjoyment, as well as in three other subscales. This leaves us with the question, if we might get a more significant result in this thesis using the same questionnaire that Xie et al. used. We will further discuss this questionnaire in one of the following chapters, but although no significant differences were found, it still acts as one example, where the questionnaire was used in conjunction with a TUI. Finally, we looked at the study by [Schneider et al., 2013] to see if a TUI can be helpful learning about more complex topics. In their case, the topic was a part taken from the vast field of neurosciences. In their study, the tangible approach indeed showed promising results, when it came to establishing a knowledge foundation. Users, who first worked with the TUI and then with textbooks, significantly outper-

Summary of the three related work studies

formed the vice versa group in two different tests, that were conducted after each treatment. This serves as an indicator, that TUIs are indeed also helpful for more complex topics and not only for simple tasks, and we will investigate, if this is also true for the abstract and algorithmic field of computer science.

To wrap up the related work chapter, we will now look at the 0/1 Knapsack problem to bridge the gap to this thesis.

2.7 The 0/1 Knapsack Problem

The Knapsack problem(KP), or more precisely the 0/1 KP, is a well-known problem in CS. It is a (weak) NP-complete problem, meaning the problem can be solved using a dynamic programming algorithm, that would require a number of steps polynomial to the size of the knapsack and the number of items (all data needs to be in integers). However, the runtime of such an algorithm would still be exponential, since the input sizes of the objects and the knapsack are logarithmic in their magnitudes. However, Garey and Johnson [1979] established that: "A pseudo-polynomial-time algorithm will display 'exponential behavior' only when confronted with instances containing 'exponentially large' numbers, which might be rare for the applications we are interested in. If so, this type of algorithm might serve our purposes almost as well as a polynomial time algorithm." This statement is true for this thesis, since we only looked at problems with small numbers, with '10' already being the largest number used, meaning that the KP serves as a complex, yet feasible problem for our purpose.

Definition:
*Dynamic
Programming*

DYNAMIC PROGRAMMING:

Method for problem solving used in math and computer science in which large problems are broken down into smaller problems. Through solving the individual smaller problems, the solution to the larger problem is discovered.

We now consider the 0/1 KP as follows:

0/1 KNAPSACK PROBLEM:

A subset of objects n has to be packed into a knapsack with capacity c . Each object o has a weight w_o and a profit value v_o . The goal of the problem is to maximize the profit-value sum of the chosen items, while not exceeding the capacity c . Each object is only allowed to be picked once or not at all, enforcing the 0/1 behavior.

Definition:
0/1 Knapsack
Problem

Henceforth, we will only consider this 0/1 version of the KP, even if we do not state it explicitly, and we will also refer to weight as cost, to profit value as rating and capacity as budget, when talking about the game implemented for this thesis.

While there are different methods to solve a KP, we will only briefly explain the already mentioned dynamic programming approach, as it was used in the software and could also be used by the participants of the study, in the pen-and-paper method.

Assume we have the following objects as shown in table 2.1, that we can place inside the knapsack which has a budget limit of 4:

Object	Rating	Cost
1	2	2
2	3	2
3	1	1

Table 2.1: Knapsack example problem.

The corresponding table used for a Dynamic Programming approach would look like this:

The first column contains the objects we already looked at so far, while the first row tells us the different budgets up to our maximum of 4. If we now add the first object into our consideration, the table will change to what can be seen in 2.3.

This information can then be used to add the second and third object into consideration. For the second object we

Object \ Budget	0	1	2	3	4
0.	0	0	0	0	0

Table 2.2: Example table used for dynamic programming approach.

Object \ Budget	0	1	2	3	4
0.	0	0	0	0	0
1.	0	0	2	2	2

Table 2.3: First step of the algorithm shown in the table.

can use the values we already got from our first step, and for the third object we can use the values from the first and second step. This allows us to get to the final table (see table 2.4).

Object \ Budget	0	1	2	3	4
0.	0	0	0	0	0
1.	0	0	2	2	2
2.	0	0	3	3	5
3.	0	1	3	4	<u>5</u>

Table 2.4: Final table containing the correct solution.

This leaves us with an optimal rating of 5 using a dynamic programming approach. For a more in-depth explanation, [Levitin, 2008] is a good starting point, especially the pages 320–322 that focus on the dynamic programming approach of solving KPs.

Chapter 3

Own work

The evaluation of the related work has shown, that TUIs have already proven to be helpful for building a solid knowledge foundation in a variety of educational fields. This motivates us, to use them not only in the practical section of CS, e.g., using them to create code for robots, but also to use them in the more abstract part of CS, that focuses on algorithmic problems. To evaluate this approach, it is compared to the pen and paper variant of solving the KP. How the TUI was implemented, how the tangibles themselves were build and how the final problems were chosen will be elaborated in the following chapter.

Overview

3.1 First Design-Implement-Analyze Cycle

In the following sections, we will discuss the first design iteration of this thesis. Following the basic principle of the Design-Implement-Analyze (DIA) cycle, we started by creating a paper-based prototype, the first tangible prototypes and the first set of KPs used in the user study. The paper and tangible prototypes went through a smaller DIA cycle over the course of this first iteration.

Overview

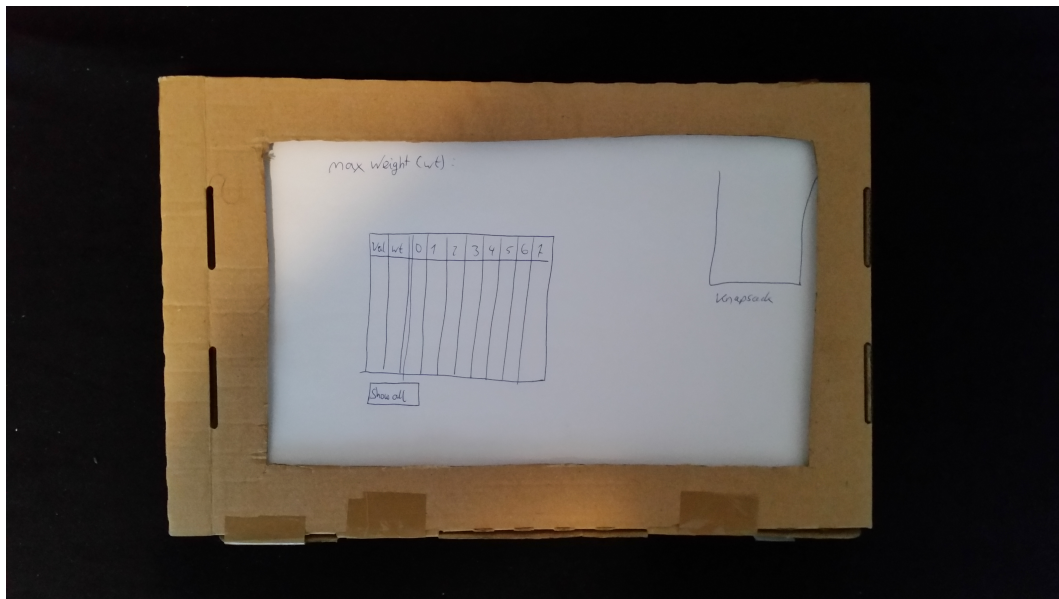


Figure 3.1: First paper-based prototype in the form of a flipbook, showing the computational table and the knapsack.

3.1.1 Flipbook and Paper Prototypes

Flipbook prototype

The first paper-based prototype build for this thesis can be seen in 3.1. It focused on the basic idea of how the software and user interaction with it could look like. The flipbook-style approach allowed for an early exploration into this user interaction, e.g., showing what would happen, if the user would put a tangible inside the virtual knapsack. In this version, the user would see a similar computational table, to what was shown at the end of chapter 2.7 “The 0/1 Knapsack Problem” in the KP explanation. This was already dropped in the first version of the software, as we wanted to focus more on enjoyment part and not on teaching the algorithm itself.

Capacity depiction of the knapsack

Building on the ideas of this first prototype, the next step was figuring out, how to best depict the capacity of the knapsack in a physical way. One idea was to go with a spatial approach. The tangibles would have different block shaped sizes, that would correspond to their weight with respect to the KP and not their real physical weight, e.g., an object that would have a weight of ‘1’ in the KP, would be



Figure 3.2: Paper prototype of a block tangible for a spatial approach.

a 10x10 block tangible (for a paper example see figure 3.2). This also led to the idea, to not use a fully virtual knapsack, but instead use a real-world box tangible.

Following the mandatory introductory talk at our chair, the spatial approach was dropped in favor of a simpler, physical-weight oriented one.

3.1.2 Building Tangible Prototypes

Following the drop of the spatial approach, the next step was to build the first tangible prototypes, that would have different weights. The first set of tangibles were build using different 8.5 cm Plexiglas circles, that were either closed or had a doughnut-shape. Stacking them together in different combinations of eight, allowed them to be filled with different amounts of bird sand, thickened with clear vanish and therefore creating different weights, while not using ecologically harmful materials such as lead(see 3.3). These first non-paper prototypes were handed out to people around our Fabrication Laboratory (FabLab), with the task to sort them by weight. This helped us to determine, that we would need roughly a 15-20g difference in weight between each tangible, to make the difference noticeable.

Figuring out
distinguishable
weight differences

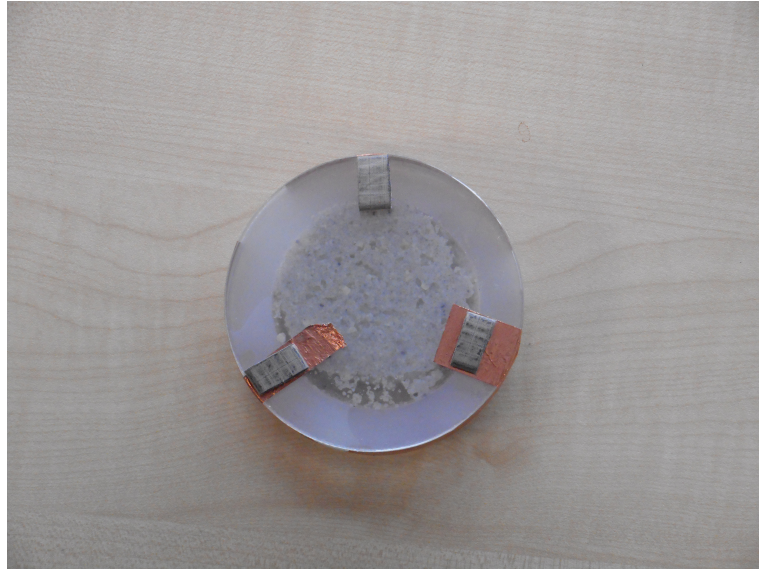


Figure 3.3: Tangible from below, showing the bird sand inside.

3.1.3 Constructing the Different Problems

Creating problems
for the TUI and pen
and paper

With the weights in place, the next focus point was designing appropriate KPs for the user study. Initially seven different problems were created for the tabletop, of which only three remained at the end, due to time constraints in the user study. Since users were not forced to take a fully algorithmic approach when working with the tangibles, three problems were designed to be easily solvable, by having an obvious distribution of items that belonged in the knapsack and of those that would not make sense. Another three problems were designed to be of intermediate difficulty. While the easy problems had objects with huge values and small weights (not physical ones, but with regard to the KP), the intermediate difficult ones consisted of a more evenly distributed set of weights and values. To finalize the set of problems, a hard one was build, that introduced an even better mix of items and several solutions close to the correct one. All problems had eight possible objects, that could be placed inside the knapsack, and every problem had a budget limit of '10'. The objects were designed to always consist of two objects with costs of '1',

another two objects with costs of '2' and then four objects with ascending costs, e.g., '3', '4', '6', '8'. At the same time, problems for the pen-and-paper part of the study were created, an easy, an intermediate and a hard one, similar to the tabletop problems, but with no special cost design in place, because it was only needed for a better weight distribution for the tangibles later on.

3.2 First Game Implementation

We will now discuss the first implementation of the game used in the user study. We will start by focusing on the graphical design and the actual goal of the game. After that, we will focus on the game implementation itself, e.g., how the game sprites were implemented, how the KPs were solved in the software or how the inputs were handled in this first iteration.

Overview

3.2.1 Graphical Design and Goal of the Game

The game takes us into a small bakery and to its master baker. The goal is to solve the KP depicted by the different pastry objects, that lay on the shelves at the start of the program (see figure 3.4). Each pastry object has a profit value in the form of a star sprite, as well as a weight in the form of a dollar sprite. The master baker has a speech bubble which contains information regarding the current state of the knapsack, including the amount of money the user has left and the current combined value of all the items inside the knapsack. To summarize this, we have sprite objects with a rating and costs, as well as a master baker who tells us the current budget we have left and the current rating of the combined objects inside the knapsack (compare with definition in 2.7 "The 0/1 Knapsack Problem")

Bakery game as a
Knapsack problem



Figure 3.4: Software after the startup. Showing the master baker, his speech bubble and the pastry objects in the shelves.

Definition:
Sprites

SPRITES:

A sprite is a type of "stand-alone" computer graphic element that has evolved along with modern computer graphics technologies. A sprite is defined as a two-dimensional image or animated image that plays a specific role, often independently manipulated, within a larger image environment. Sprites are also known as icons.

The sprites for the pastry objects were taken from [freepik](https://www.freepik.com)¹, the pastry background is a modified version of an original art by [kcstudio](http://kcstudio.in/2016/09/24/bakery/)² and finally the speech bubble was taken from [pngpik](http://www.pngpik.com/download/speech-bubble-png-image-3)³. For better visual appeal, we added a black outline on the pastry objects and mirrored the speech bubble to fit our purpose, but all credits go to the original artists. The dollar and star sprites were self-made.

¹Freepik.com

²<http://kcstudio.in/2016/09/24/bakery/>

³<http://www.pngpik.com/download/speech-bubble-png-image-3>

For the software implementation, Apple's *SpriteKit* was used as a foundation, e.g., each pastry sprite is a `KnapsackObject`, that is based on `SKSpriteNode`, while adding additional information such as: `value`, `weight`, `index` and `inSack`. These sprites get placed at the start of the scene in `setUpScene()` with their corresponding `SKLabelNode` and values determined by the `difficulty` variable that has to be manually set before the start.

To solve the KP, the user has to move the correct pastry objects onto a tray, that the master baker is holding in his hands and then press a button on his chest to see if a correct solution was found. This is then shown to the user in form of another `SKLabelNode` inside the speech bubble. The same label is used to output a message, if the user tries to use more money than available. The labels for the remaining money and the current rating of all the combined objects inside the knapsack get updated, as soon as a pastry sprite is placed onto the tray or removed from it.

3.2.2 Knapsack Solving and Input Implementation

We already established the use of *SpriteKit*. Now, we will look at how the KP solution is calculated at the start of the program, how the corresponding table is built based on the user inputs and how these inputs were handled in this first version.

On the startup of the program, two 2D-arrays get initialized by `initSolArrays()`, with the size of the knapsack in mind. This function is rather static, since we only work on KPs with an exact budget/capacity maximum of '10', meaning this would need a dynamic overhaul for future work. The `autoCalcOptValue()` is used for the calculation of the optimal solution and gets called once at the start of the program. This function uses a dynamic programming approach, working on one of the 2D-arrays, the `autoCalcArray`, that got initialized earlier.

The last entry of this array gets compared against the last entry found in the other 2D-array, the `manualCalcArray`,

Overview

Solving the knapsack problem with dynamic programming

when the user presses the button on the chest of the baker, leading to a call of the `checkSolution()` function. This user input based array gets updated in the `manualCalcSol(i:int)` function, every time a user places a `KnapsackObject` onto the rectangular hitbox of the tray. When the user removes an object from the tray, the `recalcSol()` function is called instead to correct the `manualCalcArray`.

Touch as first input
method

In the first version, the user inputs were handled in the `preProcessTraceSet` function. We did not work with tangibles yet, instead the user inputs were fully finger touch based. Every time this function got called, it would check all traces in the `Set<MTKTrace>` and if a trace was detected on a pastry object, the user could then move this object around. The distinction of the objects happened by checking the name of the `SKSpriteNode` and when the touch was lost, a dummy object was selected, that had no visual appearance and just served as a space holder until the next object was touched.

Analysis of the first
version

The first game version showed, that the basic game idea was simple enough to be picked up by multiple people of our tangible group quite fast. The problems were appropriately easy or hard, providing the user with a small challenge regarding the hard one and little to no challenge working on the easy ones. Building on these internal results, we then worked on the second version, that would implement and focus on the tangibles as the input method and finalizing our problems.

3.3 Second DIA cycle

Overview

The second DIA cycle focused on finalizing the software and therefore the tangibles, as the defining input method. We will discuss the continuous work on the tangibles based on the prototypes from the first iteration, together with problems that arose working towards their final version. While the game stayed the same, we will also discuss how the input with these newly build tangibles works, which is closely linked, to how they were build. A final discussion

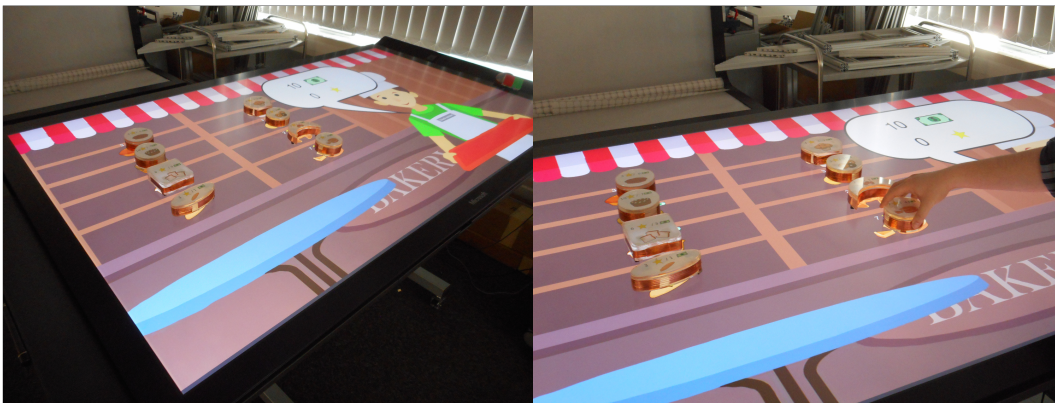


Figure 3.5: Left: Showcase of the game including the tangibles. Right: User interacting with the tangibles.

on the set problems used in the study will close this chapter and serve as the connection to the evaluation part of the thesis.

3.3.1 Building the final Tangibles

Continuing with the prototypes build in the first DIA cycle, copper foil was added around them, connecting three markers used as reference points, essentially rebuilding the PUCs by Voelker et al. [2013]. This approach led to the first issue encountered while working towards the final version of the tangibles. There was a very strict number of patterns, that would get differentiated by the MTK framework, and we hit this cap with just four different patterns. Trying to add a fifth pattern would ultimately result in a faulty recognition, where two or more objects would just jump to the same tangible, because it was mistaken for another one with a similar pattern. Thankfully, the MTK framework offered some options in the `MTKUtils`, allowing the change of certain variables linked to the recognition. With changes to:

Limited and faulty
pattern recognition

- `maxRelativeDifferenceInDistance`
- `allowedDeltaForAlphaAngle`
- `allowedDeltaForLegDistances`

it was possible to add a 5th pattern, but the faulty recognition still kept happening too often.

Solving the recognition issues by creating different shapes and modifying the PUCs approach

With this limitation in place, we could not achieve eight different tangible patterns, for the eight different pastry objects in the game, that would have the same shape. We also felt, that lowering the number of objects would make problems far too easy to solve. So we had two choices, go for the PERCs approach by Voelker et al. [2015] or further modify the existing tangibles and stick to the passive approach. We decided to stick with the PUC approach, but further improve it for our purpose. First, we set the limit of the circular tangibles to four, because this was the number where the amount of wrong pattern recognition would be low enough, after changing the `MTKUtils` values. In addition, we build four more tangibles, with different shapes close to their virtual counterpart, the muffin, the baguette, the toast and the croissant. This allowed us to create another set of different marker patterns, giving us the full set of eight different ones shown in figure 3.6. It also meant, that the shapes were closer to their virtual counterpart. This improved the recognition a lot, but sometimes the tangibles were still mistaken for another one. We could not fully fix this issue, as it would require more insight and knowledge about the MTK and the *SurfaceHub* and would most likely go beyond the scope of this thesis, that focuses more on the comparison of the tangible approach to the pen and paper method. However, we did manage to further mitigate the problem by not using a full circle of copper foil and instead, we cut it into two or three smaller parts, that would just connect to either one or two markers (see figure 3.7). This lead to earlier and faster filtering of the tangibles when they stopped moving. They also would only be fully recognized, when the user would touch all the copper foil parts, activating the connection between all three markers. Only the really "heavy" tangibles would still sometimes get recognized without the user touching them (for all final weights see table 3.1). Of course, also putting multiple tangibles too close to each other could lead to wrong recognition from time to time. This happens, because one or two markers of a certain tangible 'A' would be completed by other markers from another tangible 'B', possibly creating the pattern of 'C' when both 'A' and 'B' would not be touched correctly.

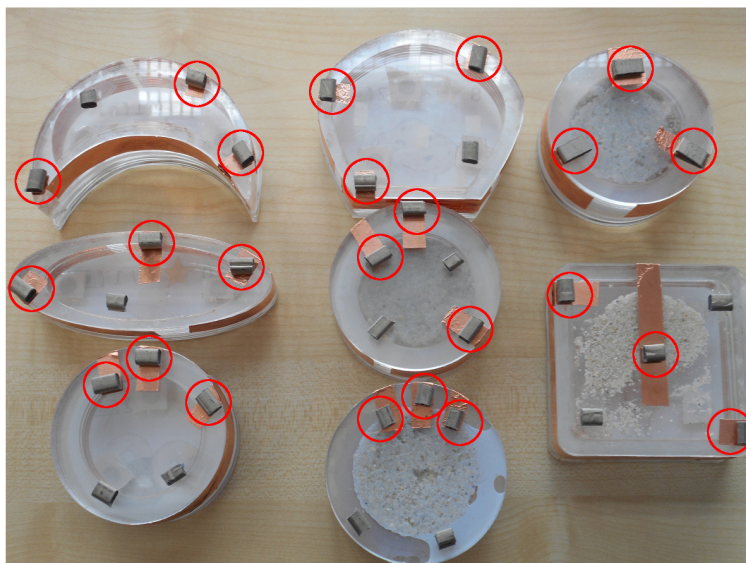


Figure 3.6: All eight tangibles with their respective markers. The red circles indicate the conductive ones. Markers without a circle are there for stabilization.

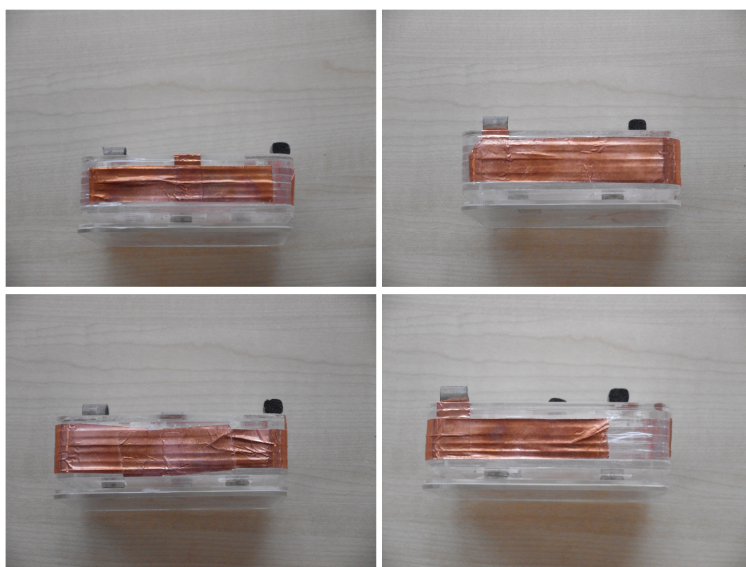


Figure 3.7: All four sides of the toast tangible showing the gaps in the copper foil, that helped to improve the PUC design for our study.

Creating the final
version

With the recognition problem solved to an acceptable degree, the last step was to make a cover that would be interchangeable for the different problems. The first idea was to simply tape sheets of paper to the tangible, so that the user can also see the value and cost of the object while looking at the sprite, but rather at the tangible itself. This idea was quickly scrapped due to not being fast and easy enough. Instead, we build multiple detachable covers from more Plexiglas. We created enough covers for the different tangibles and different problems. We then just cut the printed objects correctly and placed them in the covers that are hold together by double-sided adhesive tape. To make these covers interchangeable fast and easy, we added multiple magnets on the tangibles and the covers, so you could just switch them without much effort (see figure 3.8). This unfortunately created a gap between the cover and the tangibles since we did not place the magnets inside the tangible. In the user study we also asked the users, if this was disruptive for them or not, to see if we would need further design improvements in future work.

Object	Weight
Croissant	108
Baguette	108
Muffin	145
Doughnut	145
Toast	180
Cake1	215
Cake2	230
Cake3	251

Table 3.1: Final weights of the tangibles in grams

3.3.2 Choosing and Designing the final Problems

Choosing the final
set of problems

As we already mentioned earlier, we build multiple different problems and wanted to compare an easy, an intermediate and a hard one for both approaches against each other. While testing the system with one problem of each type it became clear, that this would bloat the user study too much, so we had to cut some time to make it feasible. Using dif-



Figure 3.8: Toast tangible with its cover attached and with its cover detached, showing the magnets.

ferent problems for each user would also have meant, that there would have been the need to create a significantly larger number of covers, which would make structuring the study worse. Therefore, we settled on only comparing an easy and a hard one against each other for each approach, where every user would get the exact same problems. From

the already created problems, two were chosen for each approach, based on the already mentioned implications on why a problem was considered easy or hard. All four problems and the two demos can be found in the Appendix B.

These tests, within our tangible group and with one outside user, also served as results for the second DIA cycle and since no further problems were found and the time was cut to roughly '40' to '60' minutes, we could transition to the actual user study, that will be discussed in the next chapter.

Chapter 4

Evaluation

In this chapter, we will discuss the design and results of the conducted user study. At the beginning of this chapter, we will explain the Intrinsic Motivation Inventory (IMI), to give the reader a basic foundation of the questionnaire used for additional user feedback. To compare a tangible approach to the manual pen-and-paper method, users are asked to solve two problems (an easy and a difficult one) with each approach. We want to analyze how participants perform with both methods and if they develop certain tactics/heuristics. It will be measured, how long it takes the user to complete the different tasks. Subsequently, we will present these results together with the additional user feedback derived from the IMI questionnaire, as well as qualitative feedback. To conclude this chapter, we will discuss design recommendations for future tangibles, that are build for this kind of problem, as well as establishing whether a tangible approach is helpful or not.

Overview

4.1 Intrinsic Motivation Inventory

The Intrinsic Motivation Inventory, or for short IMI, is a multidimensional measurement device that is used to assess "participants' subjective experience related to a target activity in laboratory experiments". The IMI consists of

Questionnaire that was used for additional user feedback

seven sub-scales and was already used in multiple laboratory experiments. We also saw Xie et al. [2008] use it in their own study that worked with a TUI. These seven sub-scales are:

- Interest/Enjoyment
- Perceived competence
- Effort
- Value/Usefulness
- Pressure/Tension
- Perceived choice
- Relatedness

The first six sub-scales were already validated and while this has not yet been done for relatedness, it does not matter, since for our purpose we settled on the predefined 22 item version of the questionnaire, that only focuses on the following four scales:

- Interest/Enjoyment
- Perceived competence
- Perceived choice
- Pressure/Tension

Perceived choice and perceived competence are theorized to be positively viewed regarding intrinsic motivation, while pressure/tension is theorized to have a negative impact. This is important for the evaluation of the questionnaire, since every question has seven possible answers ranging from "not true at all" to "very true". Every answer will therefore correspond to a number between one and seven. Some questions are asked in a way, where their score needs to be reversed to fit the scale, meaning that the actual answer will be subtracted from eight to get the correct value. The full questionnaire together with the general questions asked, can be found in the Appendix A. The

Not true at all = 1,
Very true = 7

questions that needed a reversed score, are there marked with an 'R', something that was not done for the users, but should help the reader to further understand the IMI questions from an investigator's standpoint. Some questions are also slightly changed from their original form to better fit the task, e.g., "I found the task very interesting" got changed to "I found solving Knapsack problems very interesting". This does not change the reliability or validity of these questions, but creates a bit more diversity and takes care of the otherwise monotonous style of the questions.

4.2 Hypotheses

We already mentioned in 1 "Introduction", that currently students drop out for a variety of reasons, with lack of motivation and doubts of personal qualification just being two of them. Enjoyment plays a big role in intrinsic motivation, because if a task is undesired or not enjoyable, people will find it harder to motivate themselves. Therefore, we will look if tangibles provide a more enjoyable approach to Knapsack problems, by comparing the scores of this IMI sub-scale against each other, to see if the TUI scores higher.

H1a: Tangibles raise the level of enjoyment perceived by the user, compared to working on Knapsack problems with pen and paper.

When people do not feel competent, they can also quickly become doubtful of themselves towards a specific task or topic. For that reason, we will also investigate if users give more points for the TUI in the corresponding IMI perceived competence sub-scale.

H1b: Tangibles raise perceived competence by the user, compared to working on Knapsack problems with pen and paper.

Lastly, pressure and tension, e.g., created by performance pressure or self-doubt, can often negatively impact ones performance, e.g., in sports or creative thinking ([Candle](#)

problem¹). Therefore, we will take a look if the IMI scores for the pressure/tension sub-scale are lower for the TUI, compared to the pen-and-paper method, to see if users felt less pressured working on the tasks with this new approach.

H1c: Tangibles lower the perceived pressure/tension by the user, compared to working on Knapsack problems with pen and paper.

4.3 Design of the Study

Overview

It was already mentioned in 3.3 “Second DIA cycle”, that people from our tangible research group and one outside user helped to test the software, tweak parameters and to get an estimation of how much time the study would take (approximately 40-60 minutes). In the following sections the environment, the procedure and the measurements will be discussed.

4.3.1 Environment

The study took place in a shared project space, but participants were put behind curtains while working on the pen-and-paper problems. All of the tangible tasks were performed on a *Microsoft Surface Hub 84'*, that acted as the tabletop. Its display, with a resolution of 3840x2160 pixels, can detect up to 100 touch points and the software was run on an iMac Pro in this full resolution with 30 frames per second. The touch input of the button could be executed without other hardware and the movement of the pastry objects was done through the newly build tangibles described in 3.3.1 “Building the final Tangibles”.

1	2	4	3
2	3	1	4
3	4	2	1
4	1	3	2

Table 4.1: Latin square used to determine the order of the conditions given to the user.

4.3.2 Procedure

The study focused on comparing a tangible approach to a manual pen-and-paper method for solving Knapsack problems. Since it consisted of four problems (see 3.3.2 “Choosing and Designing the final Problems”), we used a 4x4 Latin square (see table 4.1) to counterbalance learning effects. The four variables were: *Easy Tangible Problem* (1), *Difficult Tangible Problem* (2), *Easy Pen-and-paper Problem* (3) and finally *Difficult Pen-and-Paper Problem* (4).

General procedure

First, each participant was (re-)introduced to the Knapsack problem with an explanation, similar to the definition of the problem in 2.7 “The 0/1 Knapsack Problem”. It was demonstrated how the problem is solved algorithmically using pen and paper. The instructor started by explaining the different steps of the algorithm and users were asked to finish the last two rows of the example task, to see if they understood this approach. It was also mentioned that the order of the items was arbitrarily chosen, and users could change this order if they thought this could improve the algorithm and their time. After that, the instructor also showed and explained the algorithmic way of finding the correct items used in the solution. Afterwards, the participant would be taken to the table for the software demo. Again the instructor gave a short explanation on the task in the game scenery. The users were introduced to the tangibles and basic concept of putting them on the tray to “put them inside the knapsack” and using the button to check their solution. It was explained to each user, that he or she would need to touch all copper foil parts around the tangible to guarantee a flawless detection. Further advice

Demo for each method to help the user understand controls and recap on the pen-and-paper method

¹https://en.wikipedia.org/wiki/Candle_problem

was given for the differently shaped tangibles and the instructor also noted, that the tangibles had different physical weights. After the introduction, some time was given to the user to work with each tangible and get an idea of how to handle them. When the user felt, that he or she understood the goal of the game and was confident in handling the tangible input, the transition to the real problems was made. As a last remark, the participants were informed that they would have approximately '10' minutes per problem.

Two tasks for each
method

Each of the four already mentioned tasks consisted of eight objects and a knapsack with a maximum capacity of '10'. The minimum cost of an object was '1' for both approaches and the maximum was '7' for the pen and paper problems and '8' for one of the tangible problems. '1' was the minimum value of an item for both approaches and the maximum was '10'. The table for the pen-and-paper approach was printed out for the participants and the zero row used in the final steps of the algorithm was also written down already. Additionally, the easy pen-and-paper problem also had two correct solutions, based on the order of the items chosen by the user.

Solving the problems
and tolerance given

To fulfill each pen-and-paper task, the user had to find the correct maximum value and the corresponding items, using the algorithm presented earlier, but they could further improve it by changing the order of the items and using a dynamic programming approach, further referred to as cross-comparison. As they were only introduced to the general idea of checking the combination of available items in each step. For the TUI, users had to find the correct maximum value and correct set of items as well. Since the objects were also found when a correct solution was discovered, there was no need to specify them again in the tangible approach. The participants would not need to take an algorithmic approach, instead they were free on how they wanted to solve the problem, e.g., rapid testing, sorting items beforehand based on different heuristics and so on. A correct solution on the tabletop was found, when the correct items were inside the knapsack and the participant would then press the button on the chest of the baker. Although participants were told that they would be given approximately '10' minutes, the instructor would extend this by 2-3 minutes, if a

user was close to finishing the task, e.g., filling out the last table row, already backtracking the correct items in the pen-and-paper approach or close to finding the correct solution on the tabletop.

The users were given the choice to take a small break between each pen-and-paper task, if they were in a group that based on the Latin square would do both problems one after the other. In addition participants had a small break if they needed to switch to the tabletop or if both tabletop problems also occurred after each other, since the software had to be restarted for the correct log files and the instructor had to change the covers of the tangibles to the correct problem, while also setting them up again at their original location on the table.

Short breaks
between the task

4.3.3 Measurements

For the pen-and-paper approach, the time started when the instructor gave the start signal to the participant and it was stopped when the user felt he did everything correct and put the cap back on the pen. The completed tables for the easy and difficult problem, were later on corrected in a similar way of how it would have been done with exam tasks of the same type. Calculation errors and algorithmic errors were counted as one error as long as it was obvious that it was a follow-up error that was based of an earlier mistake, e.g., continuing correct calculations based on a wrong one, mistaking costs for value, or while using the cross-comparison method, slipping into the wrong column or row.

Pen-and-paper
logging included the
time it took until
completion, as well
as a correction of the
solution at a later
stage

Time measuring started after the startup of the software was finished, when the user could start to move the tangibles, and it was stopped when the user put the correct combination of items on the tray and hit the "Check solution" button. For one user the tabletop lost connection due to a malfunction of the HDMI cable to the iMac, while working on one of the problems. The times for this one specific user were taken out of the evaluation. Additionally, it was logged how often the user would enter a wrong solution,

Logging for the TUI
included the time and
additional information
about the game and
tangible state

meaning he or she pressed the button with the wrong set of items on the tray, as well as if the correct solution was found at all.

Questionnaire for demographic information, tangible design, prior knowledge, problem comparison and user ratings

After all four tasks were done, each user was given a questionnaire with demographic questions, a question if they would see themselves as experts solving algorithmic problems, multiple questions about the design of the tangibles, e.g., if the different shapes were disruptive or not, a question regarding the different difficulties of the problems, a preference question and the set of 22 questions taken from the IMI questionnaire itself. They could also give personal comments at the end of the study. The full questionnaire can be seen in the Appendix A.

4.4 Participants

16 people (aged from 16 to 28, $M = 23$, $SD = 2.7$, 5 female) participated in the study. 14 people were familiar with the problem and the algorithm itself, since they were CS students. Most of these 14 persons finished their bachelor degree already and are currently enlisted as master students at our university. The two people without prior knowledge were the youngest participant, who still goes to school and was at our chair for an internship and another intern, who is currently studying a subject close to CS. Of the 16 participants only 1 had worked with a TUI before this study.

4.5 Results

Data summary

Combined the 16 participants performed 64 tasks. Only one participant took too long for one pen-and-paper problem and had to be stopped, even after he was given some extra time. For this same user the table lost connection to the iMac, therefore all of his times were excluded from the data analysis, as it was mentioned earlier already. The 16 participants answered 352 IMI questions and 96 additional written questions, together with many more spoken comments

and answers to questions while or shortly after working on the different tasks. The results of this data will be presented and discussed in the following sections.

4.5.1 Quantitative

The data of the IMI questionnaire and the times were mostly normally distributed, except for the tangible and pen-and-paper data for the choice sub-scale, the tangible data for the tension sub-scale and the times for the hard tangible and easy pen-and-paper problem. This was tested with Shapiro-Wilk tests with 95% confidence intervals.

Data distribution

IMI Questionnaire Rating

We analyze the answers given in the IMI questionnaire to assess user's self-reported intrinsic motivation working with each approach.

Self reported enjoyment and competence scored significantly higher, tension scored significantly lower and choice had no significant differences

Running a Wilcoxon signed-rank test revealed significant differences for the enjoyment, competence and tension sub-scale and no significant difference for the choice one. The 'Z' and 'p' values of the test can be seen in 4.2.

Sub-scales \ Parameters	Z	p
Enjoyment	4.07564	< 0.0001
Competence	3.22951	0.0012
Choice	1.27550	0.2021
Tension	-3.03380	0.0024

Table 4.2: Overview of the 'Z' and 'p' values of the Wilcoxon signed-rank test.

Looking at each sub-scale individually, the users overall gave the tangible approach higher scores in the enjoyment (M = 5.8, SD = 0.78 compared to M = 3.5, SD = 1.32), competence (M = 5.4, SD = 1.10 compared to M = 3.8, SD = 1.33) and choice (M = 5.9, SD = 0.97 compared to M = 5.0, SD =

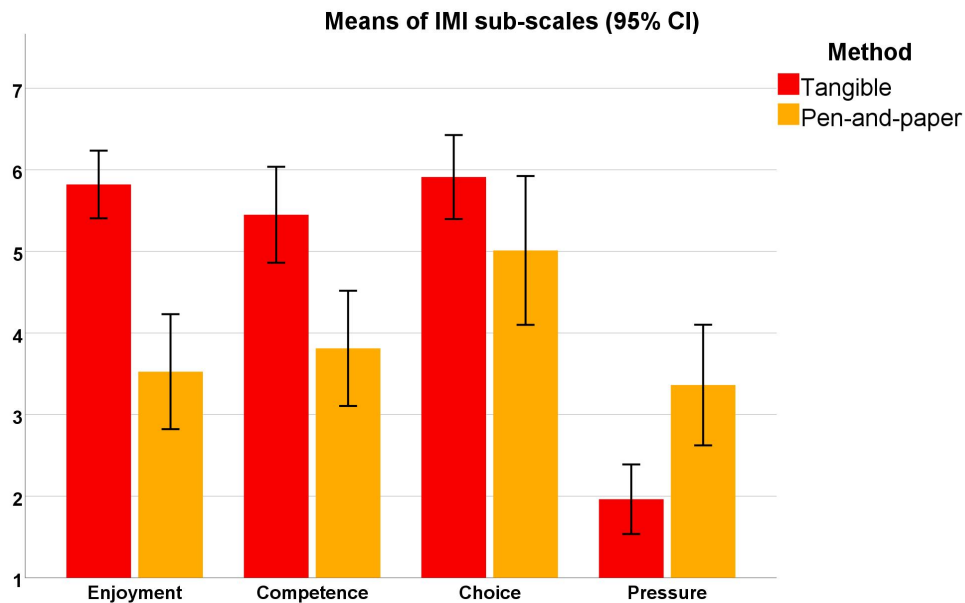


Figure 4.1: Mean scores of the IMI questionnaires sub-scales with 95% confidence intervals.

1.71) section, while giving a lower score for the perceived tension ($M = 1.9$, $SD = 0.80$ compared to $M = 3.5$, $SD = 1.3$). The visual depiction of the mean values of each sub-scale and for each approach can be seen in figure 4.1.

Time to solve

Tangible approach was significantly faster, independent of the difficulty

The time to solve (TTS) is analyzed to get insight in the participants overall performance and it also serves as another point for further intrinsic motivation analysis. A pooled t-test revealed significant differences for the two approaches for the easy problem ($t[28] = -11.0154$, $p < 0.0001$) and also for the hard one ($t[28] = -9.94952$, $p < 0.0001$). However, comparing the times of the tangible approach for the easy and hard problem showed no significant difference ($p = 0.697$), this was the same when comparing the times of the easy and hard pen-and-paper problem ($p = 0.110$).

Participants took the least amount of time to solve (in seconds) for the easy problem with tangibles ($M = 105$, $SD =$

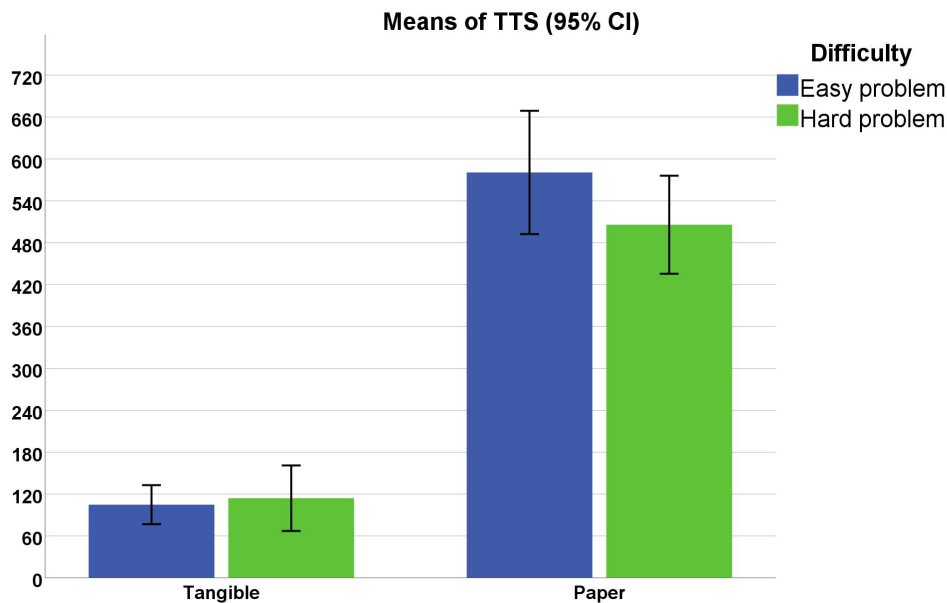


Figure 4.2: Means of TTS (seconds) for both methods and both difficulties with 95% confidence intervals.

51), and the longest time to solve the easy problem with pen-and-paper ($M = 581$, $SD = 160$). The TTS for each approach and for each problem can be seen in figure 4.2.

Additional design questions

Before the IMI part of the questionnaire, users were asked several questions regarding the design of the tangibles and design of the problems. Users were asked if the physical weight difference was detectable for them (Yes = 9, No = 3, Not sure = 4), if they felt that the difficulty was the same for both methods (Yes = 8, No = 6, Not sure = 2), if the different shapes were disruptive (Yes = 1, No = 15) and if the gap created by the detachable covers was disruptive (Yes = 0, No = 16). In addition, nearly all participants preferred the tangible over then pen-and-paper approach (Tangible = 13, Pen-and-paper = 2, No preference = 1).

Weight difference was not really detectable, mixed results regarding the difficulty, close to none problems with the tangibles and large preference towards the TUI approach

4.5.2 Qualitative

Besides the quantitative results, participants were given the option to write down their own personal comments at the end of the questionnaire. Additionally, the instructor noted down comments made during the study together with specific questions asked, if something was unclear to either the user or the instructor, e.g., on why and how a participant would change the order of the items in the pen-and-paper approach.

Feedback for the TUI
approach

While working with the TUI all participants mentioned that they would compare the value they get with the costs of the item. While doing so, only 2 users would actively presort the tangibles on the table and the rest would do it thinking aloud, but without physically moving the tangibles. 6 participants actively reported that they enjoyed working with the TUI and another 5 mentioned this after finishing the tasks. 12 participants used a combination of dragging the tangible and lifting them up, while 3 stuck to lifting and setting them down and 1 user only dragged them across the screen. The user who was familiar with tangibles also noted down, that the tangibles were made pretty good and that he noticed no problems working with them. Additionally, users had a tendency to re-evaluate their solution before clicking the button to actually check it automatically, this was done by 11 participants.

Feedback for the
pen-and-paper
approach

5 users reported that they felt like they were back at an exam and 6 additional ones reported, that they struggled with the mental arithmetic required. Only 7 people used the cross-comparison method, that the dynamic programming approach suggests. Just 5 people rearranged the objects in the pen-and-paper approach, although every user was explicitly told, that the given order of the items is arbitrarily designed and that they are free to choose a different order. 2 people performed the mental arithmetic loudly and the other 14 stuck to silently calculating the values. The 2 participants without prior knowledge still performed as well as the other ones.

4.6 Discussion

In this section, we now will summarize and discuss the quantitative and qualitative results. We will do so by looking at the benefits each method provides.

4.6.1 Pen-and-paper Approach

The intrinsic motivation was significantly lower using the pen-and-paper approach to solve KPs. Participants reported, that they had less fun working on the tasks, while also feeling less competent and more pressured. These are all things negatively impacting ones intrinsic motivation and could lead, e.g., to frustration or lack of motivation. As we discussed in 1 “Introduction”, students currently struggle with said motivation and self-doubt about their performance, leading to a potential withdrawal from their studies.

Lower intrinsic
motivation

It took participants significantly longer to fully solve the different problems with pen-and-paper. This was somewhat to be expected, because mental arithmetic takes time to be correct and users had to do all steps of the algorithm correctly. The uncertainty, if a found solution is correct and if all the steps had be done correctly lead to further stress and lack of confidence, as we saw in the comments made after finishing the tasks. This is partially caused by the lack of feedback, while working on the problem. The longer time to solve most likely also had a negative impact on the enjoyment IMI scale, since tasks that are perceived as undesired are often already viewed negatively in general and when it takes longer to do these tasks, this negative reluctance gets amplified.

Steady but slow

Despite this negative view on the approach, it is important to note that it remains the validated correct algorithmic way of solving KPs with pen-and-paper. If no mistakes are made, then the globally correct solution will always be found. But only a handful of the participants applied heuristics working with pen-and-paper, to simplify

Remains the fully
correct solution

and streamline the algorithmic method.

4.6.2 Tangible Approach

Higher intrinsic
motivation

Using the tangible approach proved a significantly higher intrinsic motivation. Participants reported, that they enjoyed working on the problems in the questionnaire and by their spoken comments. They also felt more competent and less pressured. These are all positive things regarding the intrinsic motivation and by raising it, we might be able to ease the lack of motivation and self-doubt that students currently have.

Faster method of
solving KPs

Participants were significantly faster using the tangibles to solve a given problem. Again this was somewhat to be expected, because only simple calculations had to be done and users with a keen eye could spot the correct solution relatively fast. The direct feedback via the speech bubble helped the users, even when they did not instantly find the correct solution. No participant was demotivated, instead most cheered themselves on that they would most likely be close to the correct solution already. This was also reflected in their self reported perceived competence, which was higher for the TUI. The lower time to solve most likely also had a positive impact on the perceived pressure, since users would know that they would easily stay below the given time limit and therefore easing the fear of being too slow. Additionally, while the times were faster, users also valued to find the correct solution above being a little bit faster most of the time. This was indicated by the re-evaluation of the solution before clicking the button as it was already mentioned earlier.

TUI helped to find a
heuristic

As we saw in the qualitative feedback, every user would build a weight/cost to value ratio. This is a common approach to approximate the correct solution of KPs, as it can help to speed up the process of finding a correct solution. Of course this solution is only a local one, but often times it coincides with the global one as well. While using this approach with the TUI showed good results for everyone, it was something that most users did not use working with

the pen-and-paper approach. This might be an indicator that a TUI can help to find or establish certain heuristics.

4.7 Design Recommendations

Using the quantitative and qualitative data derived from the study, we now can give some design recommendations on what worked already and what might need some further improvements.

The problems for this study, and their respective difficulty, were based on the instructors subjective impression, paired with the reports of only very few people from the pre-test of the software. The times to solve and the user reports on the difficulty indicate, that this subjective impression might have been skewed. For further research the problems would need to get re-evaluated if and why they are easy or hard to solve.

Re-evaluate the problems

The reports if the weight differences were noticeable or not were mixed. Either the weight differences were still too small or the users preferred way of moving the tangibles around, drag them or lift them, would prevent them from noticing and utilizing it. We would suggest to run a better pre-study next time, that fully focuses on the perceivable weight differences. We also suggest to then look more closely at what movement users would perform, to see if dragging the tangibles would also mean that users would have a harder time detecting the weight differences.

Increase the weight differences

User reports showed, that neither the gap nor the different shapes were disruptive. We therefore conclude, that users were overall satisfied with the design of the tangibles and that they had no problem to touch them correctly. We still would suggest building another set of tangibles with the magnets inside, to see if then the users would prefer this gapless version. Despite the good design, it would still be important to further improve the patterns and pattern recognition, to hopefully enable the use of more passive tangibles.

Tangible already quite good

Chapter 5

Summary and future work

This final chapter concludes this thesis. We will summarize our work on a tangible approach to work on KPs. Furthermore, we will sketch out future work as this new approach serves as an entry point, that will most likely open up many more potential research topics.

5.1 Summary and contributions

Tangibles have already been used in a learning environment to teach specific topics or help with building a knowledge foundation, as we saw in 2 “Related Work”. With our work we focused on creating a TUI that works on multi-touch tabletops specifically and is not based on a camera vision approach. We also aimed to build a system that focuses on the abstract part of CS, as most systems either focus on a completely different subject or on programming, when it comes to CS. Therefore, the KP seemed to be a good fit, since it is a famous abstract problem, that is easy to understand while yet providing a certain challenge solving it.

We aimed to create a TUI for one abstract CS problem

For this thesis, we therefore programmed our own small game. We described the design process behind the game

We implemented our own small game

starting with the flipbook mentioned in 3.1.1 “Flipbook and Paper Prototypes”, going over a first implementation in 3.2 “First Game Implementation”, to arriving at 3.3 “Second DIA cycle”. We set a bakery as the stage of this game with multiple pastry objects representing the objects of the KP. A set of tasks was created for the game, of which three remained for the tangible approach and three remained for the pen-and-paper one. We discussed how these problem were constructed and why they would be considered to be easy or hard. Additionally, we described how we used and improved the existing MTK framework to help us detect our eight different tangibles in 3.3.1 “Building the final Tangibles”.

We build our own set of passive tangibles

To make this TUI approach possible we had to build our very own set of tangible based on the PUCs. We modified this base by not using a full copper mantle. Instead our tangibles would only get recognized when the user would touch all two or three different copper foil parts. Again we discussed the design process of these new tangibles, that would also have different physical weights as a depiction of their costs in the KP. Finally, in 3.3.1 “Building the final Tangibles” we described their latest version which would also feature a detachable cover using magnets.

Our evaluation helped to detect the intrinsic motivation and tactics of the users

In our user study, which we described in 4 “Evaluation”, participants had to solve four different KPs, two with pen-and-paper and two with the TUI. We evaluated quantitative (IMI, general design questions, time to solve) and qualitative (user comments at the end of the questionnaire, comments made during or shortly after the tasks) feedback. This evaluation showed a significant improvement of the intrinsic motivation that user perceived, while also allowing them to solve the problems faster. Additionally users applied a weight to value heuristic using the tangibles, something most participants did not do working with pen-and-paper. They also did not solve the problems by just rapidly and randomly testing solutions, instead they even re-evaluated theirs before using the button to check it automatically. All this data was presented and discussed in 4.5 “Results”, and it allowed us to derive some design recommendations, such as restructuring the problems and increasing the weight difference between the tangibles.

This work contributes to the research on TUI in the field of learning and teaching, providing an entry point for further research regarding abstract CS problems and algorithms. The proposed tangible approach proved to be significantly better with regard to intrinsic motivation, something that could help to lower the number of dropouts in CS studies. Additionally, the tangible method allowed users to quickly spot one heuristic that can be used to approximate the solution of a KP. As this thesis serves an entry point, there are many interesting research topics linked to it, some of which we will now discuss.

Contribution of the thesis

5.2 Future work

First, the times to solve and personal user feedback indicated, that the difficulty of the problems might have been skewed. Therefore, we would suggest to re-evaluate why a problem is difficult or not, this also includes but is not limited to increasing the number of tangibles and therefore the number of pastry objects. To do this, further improvement in the MTK is needed, to allow a better recognition of more passive tangibles, otherwise a swap to active tangibles might be another solution, as they would also provide even more feedback, e.g., LED and vibration.

Scale up and re-evaluate the problems, further improve MTK framework

After tweaking the initial study, we would suggest running a user study with first or pre-study participants. We only had two people without prior knowledge to the problem and algorithm, yet they did not perform significantly worse than the other ones. Therefore, we think it would be interesting to test the system in a workshop-like environment, as it is currently done for programming as a pre-first-semester course.

Run the tweaked study with first semester students

Compare it with a
digital approach

Based on the results of the qualitative data and the reported high value of the feedback provided by the TUI, we would also suggest to compare it not to a physical pen-and-paper approach. Instead it would be interesting comparing the TUI against another digital approach, where users would fill out the table with mouse and keyboard, while the software would give them error feedback, which might improve their perceived competence and performance in general.

Appendix A

User Study Consent Form and Questionnaire

The following consent form and questionnaire was handed out to the participants of the study. The consent form was filled out before the study started and the questionnaire was handed out after finishing all the tasks.

Informed Consent Form

Evaluating Tangibles on Tabletops to solve Knapsack problems.

PRINCIPAL INVESTIGATOR Marvin Bruna
Media Computing Group
RWTH Aachen University
Phone: 01525/1908025
Email: marvin.bruna@rwth-aachen.de

Purpose of the study: The goal of this study is to assess if Tangibles help solving the 0/1 Knapsack problem. Users will be asked to solve different Knapsack problems with Tangibles as well as with pen and paper. They will also be asked to fill out a questionnaire after working on all of the problems. I will use these questionnaires as well as the time and the amount of errors that the users make as data points.

Procedure: Participation in this study will involve two phases. In the first phase, you will be introduced to the Knapsack algorithm with pen and paper and to the software using Tangibles. In the second phase, you will be asked to solve four different Knapsack problems with different difficulties using both methods. This study should take about 40-60 minutes to complete.

After the study, we will ask you to fill out a questionnaire about how you feel towards both methods, together with some demographic questions.

Risks/Discomfort: You may become fatigued during the course of your participation in the study. There are no other risks associated with participation in the study. Should completion of either the tasks or the questionnaire become distressing to you, the study will be terminated immediately.

Benefits: The results of this study will be useful for further work regarding the usage of Tangibles in a learning and computer science environment.

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

Cost and Compensation: Participation in this study will involve no cost to you. There will be snacks and if you feel thirsty during or after the participation, there will also be a drink for you.

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 Marvin Bruna at 01525/1908025 email: marvin.bruna@rwth-aachen.de

ID:

General questions:

How old are you: _____

Gender: _____

Do you consider yourself an expert solving algorithmic problems?: Yes No

Do you think the weight differences were distinguishable enough?

Yes No I am not sure

Do you think the difficulty of the problems was the same for pen and paper and the Tangibles?

Yes No I am not sure

Was it disruptive that some Tangibles had different shapes than the others?

Yes No I am not sure

Was it disruptive that there was a gap between the Tangible and the detachable cover?

Yes No I am not sure

What method did you prefer?

Tangibles Pen and paper No preference

For each of the following statements, please indicate how true it is for you, using the following scale:

Not true at all			Some what true			Very true
1	2	3	4	5	6	7

1. While I was working on the problems I was thinking about how much i enjoyed it.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

2. I did not feel nervous at all working on the tasks. (R)

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

3. I felt that it was my choice to do the tasks.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Ver y true
1	2	3	4	5	6	7

4. I think I am pretty good at solving Knapsack problems.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Ver y true
1	2	3	4	5	6	7

5. I found solving Knapsack problems very interesting.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Ver y true
1	2	3	4	5	6	7

ID:

6. I felt tense while doing the tasks.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

7. I think I did pretty well solving the Knapsack problems, compared to other users.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

8. Doing the tasks with this method was fun.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

9. I felt relaxed while solving the problems. (R)

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

10. I enjoyed doing the tasks very much.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

11. I didn't really have a choice about doing the tasks. (R)

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

12. I am satisfied with my performance solving the problems.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

13. I was anxious while working on the problems.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

14. I thought working on the tasks was very boring. (R)

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

15. I felt like I was doing what i wanted to do while I was working on the problems.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

16. I felt pretty skilled using this method to solve Knapsack problems.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

17. I thought working on the tasks was very interesting.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

ID:

18. I felt pressured while doing the tasks.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

19. I felt like I had to do the tasks. (R)

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

20. I would describe the method as very enjoyable.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

21. I did the tasks because I had no choice. (R)

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

22. After working on the different problems for awhile, I felt pretty competent.

with Tangibles						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

with pen & paper						
Not true at all			Some what true			Very true
1	2	3	4	5	6	7

Additional comments:

Appendix B

Problems used in the User Study

Here, we present the different problems used in the user study. Starting with the demo of the TUI, followed up by the easy tangible, the hard tangible, the demo pen-and-paper, the easy pen-and-paper and finally the hard pen-and-paper problem.



10
0

- 3 ★ / 5 €
- 10 ★ / 7 €
- 4 ★ / 2 €
- 3 ★ / 2 €
- 2 ★ / 1 €
- 7 ★ / 4 €
- 3 ★ / 1 €
- 6 ★ / 3 €

Check solution

BAKERY



10
0



8 ★ / 2 €



5 ★ / 2 €



7 ★ / 1 €



7 ★ / 4 €



3 ★ / 5 €



9 ★ / 6 €



6 ★ / 3 €



10 ★ / 1 €



BAKERY

Check solution



10
0

- 2 ★ / 2 €
- 4 ★ / 2 €
- 1 ★ / 1 €
- 5 ★ / 5 €
- 8 ★ / 6 €
- 10 ★ / 8 €
- 5 ★ / 4 €
- 3 ★ / 1 €

Bibliography

Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979. ISBN 0716710447.

Heublein and Schmelzer. Die entwicklung der studienabbruchquoten an den deutschen hochschulen. https://www.dzhw.eu/pdf/21/studienabbruchquoten_absolventen_2016.pdf, 2018. [Online; accessed 12-April-2019].

Heublein, Schmelzer, Sommer, and Wank. Die entwicklung der schwund-und studienabbruchquoten an den deutschen hochschulen. https://www.dzhw.eu/pdf/21/his-projektbericht-studienabbruch_2.pdf, 05/2008. [Online; accessed 12-April-2019].

Heublein, Ebert, Hutzsch, Isleib, König, Richter, and Woisch. Motive und ursachen des studienabbruchs an baden-württembergischen hochschulen und beruflicherverbleib der studienabbrecherinnen und studienabbrecher. https://www.dzhw.eu/pdf/21/BaWue_Bericht_gesamt.pdf, 2017. [Online, accessed 12-April-2019].

Michael S. Horn and Robert J. K. Jacob. Tangible programming in the classroom with tern. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '07, pages 1965–1970, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-642-4. doi: 10.1145/1240866.1240933. URL <http://doi.acm.org/10.1145/1240866.1240933>.

Michael S. Horn, R. Jordan Crouser, and Marina U.

- Bers. Tangible interaction and learning: The case for a hybrid approach. *Personal Ubiquitous Comput.*, 16(4):379–389, April 2012. ISSN 1617-4909. doi: 10.1007/s00779-011-0404-2. URL <https://doi.org/10.1007/s00779-011-0404-2>.
- Tamara M Lackner, Kelly Dobson, Roy Rodenstein, and Luke Weisman. Sensory puzzles. In *CHI'99 Extended Abstracts on Human Factors in Computing Systems*, pages 270–271. ACM, 1999.
- Anany Levitin. *Introduction To Design And Analysis Of Algorithms, 2/E*. Pearson Education India, 2008.
- Rene Linden. Multitouchkit: A software framework for touch input and tangibles on tabletops and. Master's thesis, RWTH Aachen University, Aachen, September 2015.
- Fred Martin and Mitchel Resnick. Lego/logo and electronic bricks: Creating a scienceland for children. *Advanced Educational Technologies for Mathematics and Science*, 107:61, 1993.
- Timothy S. McNerney. From turtles to tangible programming bricks: Explorations in physical language design. *Personal Ubiquitous Comput.*, 8(5):326–337, September 2004. ISSN 1617-4909. doi: 10.1007/s00779-004-0295-6. URL <http://dx.doi.org/10.1007/s00779-004-0295-6>.
- Rodrigo Mesquita. Seymour papert. <https://www.flickr.com/photos/rmesquita/2544780949/>, June 30, 2005. [Online; accessed 04-April-2019].
- Radia Perlman. Using computer technology to provide a creative learning environment for preschool children. 1976.
- Bertrand Schneider, Jenelle Wallace, Paulo Blikstein, and Roy Pea. Preparing for future learning with a tangible user interface: the case of neuroscience. *IEEE Transactions on Learning Technologies*, 6(2):117–129, 2013.
- Statistisches Bundesamt. Studierende und studienanfänger/-innen im 1. hochschulsemester nach fächergruppen, hochschularten und geschlecht.

<http://www.datenportal.bmbf.de/portal/de/Tabelle-2.5.21.html>, 01/2019. [Online; accessed 12-April,2019; Datenlizenz Deutschland – Namensnennung – Version 2.0 (<https://www.govdata.de/dl-de/by-2-0>).

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 *ITS '13: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 101–104, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2271-3. doi: 10.1145/2508468.2514926.

Simon Voelker, Christian Cherek, Jan Thar, Thorsten Karer, Christian Thoresen, Kjell Ivar Øvergård, and Jan Borchers. Percs: Persistently trackable tangibles on capacitive multi-touch displays. In *UIST '15: Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, UIST '15, pages 351–356, New York, NY, USA, November 2015. ACM. doi: 10.1145/2807442.2807466. URL <http://dx.doi.org/10.1145/2807442.2807466>.

Lesley Xie, Alissa N. Antle, and Nima Motamedi. Are tangibles more fun?: Comparing children’s enjoyment and engagement using physical, graphical and tangible user interfaces. In *Proceedings of the 2Nd International Conference on Tangible and Embedded Interaction*, TEI '08, pages 191–198, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-004-3. doi: 10.1145/1347390.1347433. URL <http://doi.acm.org/10.1145/1347390.1347433>.

Index

0/1 Knapsack Problem, 16–17
- Definition, 16–17

BrainExplorer, 13–14
Button Box, 7–9

DIA cycle, 19
Dynamic Programming, 16

EventTable, 12–13

Intrinsic Motivation Inventory, 33–35, 41–42

LEGO Mindstorms, 9–10
Logo, 7

MultiTouchKit, 5–6
Music Blocks, 10

Pooled t-test, 42–43

Saphiro-Wilk, 41
Sensory Puzzles, 10
Slot Machine, 7–9
SpriteKit, 25
Sprites Definition, 23–24

Tangible constructions, 6–7
- PERCs, 7
- PUCs, 6

Tern, 11–12

Wilcoxon signed-rank test, 41–42

