

Institut für Visualisierung und Interaktive Systeme (VIS)
Abteilung Mensch-Computer-Interaktion (MCI)
Universität Stuttgart
Universitätsstraße 38
D – 70569 Stuttgart

Diplomarbeit Nr. 3517
Interaktionskonzept zur
Datenanalyse auf Basis eines
Tabletop-Computers und
Tangible-Objekten

Alexander Duschek

Studiengang:	Softwaretechnik
Prüfer:	Prof. Dr. Albrecht Schmidt
Betreuer:	M. Sc. Thomas Kubitza Dipl.-Phys. Michael Raschke
begonnen am:	27.05.2013
beendet am:	26.11.2013
CR-Klassifikation:	H.5.2, H.5.3, I.3.6

Kurzfassung

Neue Interaktionskonzepte basieren auf Tangible User Interfaces oder Finger- und Gestensteuerung. Sie ermöglichen kollaboratives Arbeiten, natürlichere Interaktion und machen mehr Spaß. Die kollaborative Datenanalyse lässt sich mit traditionellen Eingabegeräten wie Maus und Tastatur nur schwer durchführen. Daher stellt diese Diplomarbeit ein Interaktionskonzept zur Datenanalyse auf Basis eines Tabletop-Computers und Tangible-Objekten vor. Die Tangibles verfügen dabei über Sensoren und Kommunikationsgeräte (*Active Tangibles*), um neue Interaktionsmodalitäten anzubieten und die Benutzererfahrung zu verbessern. Das Konzept sowie der entstandene Prototyp wurden jeweils an Hand einer Studie evaluiert.

Abstract

New interaction techniques can be based on Tangible User Interfaces or finger and gesture control. These types of interfaces allow users to collaboratively work in a team with natural interactions and to have fun. Due to that a collaborative data analysis is difficult to realize with standard input devices like mice and keyboards, this work presents an interaction concept for data analysis based on a tabletop computer and tangible objects. The tangible objects have sensors and communication modules (*Active Tangibles*) to offer new interaction modalities and enhance user experience. The concept and the developed prototype were both evaluated by a user study.

1 Inhaltsverzeichnis

Kurzfassung	2
Abstract	2
1 Einleitung	9
1.1 Aufbau der Arbeit	9
2 Grundlagen.....	11
2.1 Natural User Interfaces.....	13
2.2 Tangible User Interfaces.....	14
2.3 Tabletops	16
2.3.1 Eingabetechniken und Funktionsweise.....	16
2.3.2 Beispiele	20
2.4 Techniken zur Entwicklung von Tangibles.....	21
2.4.1 Einführung in Rapid-Prototyping.....	21
2.4.2 Rapid-Prototyping-Verfahren.....	21
3 Verwandte Arbeiten.....	24
4 Aufgabenbeschreibung	30
4.1 Hintergrund	30
4.2 Aufgabenbeschreibung.....	30
4.3 Vorgehensweise.....	30
5 Anforderungsanalyse	32
5.1 Anforderungen an das Konzept.....	32
5.2 Anforderungen an die Darstellung	33
5.3 Anforderungen an die Tangibles	35
5.4 Sonstige Anforderungen.....	37
6 Lösungskonzept.....	39
6.1 Vorüberlegungen	39
6.2 Vorstudie	43
6.2.1 Durchführung	48
6.2.2 Zusammenfassung der Studienergebnisse	50
6.2.3 Einfluss der Ergebnisse auf das Konzept	53
6.3 Interaktionskonzept des Prototyps	54
6.3.1 <i>Typen von Tangibles</i>	54
6.3.2 <i>Startzustand des Interaktionskonzepts</i>	54
6.3.3 <i>Erstes Tangible wird platziert</i>	55
6.3.4 <i>Ändern und Löschen von Zuweisungen</i>	56
6.3.5 <i>Farben</i>	56

6.3.6	<i>Verkettung mehrerer Tangibles</i>	56
6.3.7	<i>Visualisierungen anzeigen</i>	57
6.3.8	<i>Filtern und Hervorheben</i>	58
6.3.9	<i>Externen Bildschirm einbeziehen</i>	59
7	Prototyp	60
7.1	Tangibles	60
7.1.1	Funktionsweise	60
7.1.2	Entwicklung	60
7.1.3	Tangible-Software	64
7.2	Tabletop-Software	67
7.2.1	Modell der Eye-Tracking-Daten	67
7.2.2	Architekturüberblick	69
7.2.3	Benutzerschnittstelle	69
7.2.4	Kommunikation	72
7.2.5	Graphstruktur und Datenweiterleitung	73
7.2.6	Datenspeicherung	76
8	Prototyp Evaluation	77
8.1	Durchführung	77
8.2	Zusammenfassung der Studienergebnisse	85
8.3	Schlussfolgerungen	86
8.4	Nachträgliche Änderungen am Prototyp	86
9	Zusammenfassung	88
9.1	Bewertung	88
10	Ausblick	90
	Anhang A: Startfragebogen der Vorstudie	93
	Anhang B: Bewertungsbogen der Vorstudie	95
	Anhang C: Bewertungsbogen NASA-TLX der Evaluationsstudie	96
	Anhang D: Vergleichsfragebogen der Evaluationsstudie	97
	Literaturverzeichnis	98
	Erklärung	100

Abbildungsverzeichnis

Abbildung 1 – Entwicklungen im Bereich NUI, TUI und Tabletop-Computern	12
Abbildung 2 – Beispiele für Natural User Interfaces	13
Abbildung 3 – Great Dome (MIT)	14
Abbildung 4 – Shakepad.....	15
Abbildung 5 – Sifteo Cubes.....	15
Abbildung 6 – Kapazitative Techniken im Vergleich.....	17
Abbildung 7 – Schematische Darstellung der Frustrated Total Internal Reflection-Technik (FTIR).....	17
Abbildung 8 – Schematische Darstellung der Diffuse Illumination-Technik (DI)	18
Abbildung 9 – Schematischer Aufbau der Diffused Surface Illumination-Technik (DSI).....	19
Abbildung 10 – Schematischer Aufbau der PixelSense-Technik	19
Abbildung 11 – Tabletop-Computer PixelSense der ersten Generation von Microsoft	20
Abbildung 12 – Natural Interaction SensitiveTable als MediaBrowser	21
Abbildung 13 – Bild eines 3D-Druckers.....	22
Abbildung 14 – Arduino Mainboard.....	23
Abbildung 15 – Beispiel einer Digitalkamera mit dem .NET Gadgeteer Kit.....	23
Abbildung 16 – Das linke Bild zeigt die Benutzeroberfläche von VPlay.....	24
Abbildung 17 – Tangible für VPlay	24
Abbildung 18 – Kollaboratives Arbeiten auf dem ReactTable	25
Abbildung 19 – Marker-Typen des ReactTables	25
Abbildung 20 – Aufbau des SenseTable-Systems	26
Abbildung 21 – SenseTable Pucks	26
Abbildung 22 – Powerwall Interactions	27
Abbildung 23 – FacetStreams.....	28
Abbildung 24 – Facet Token und Result Token.....	28
Abbildung 25 – Skizzen der drei geplanten Tangible-Typen.....	40
Abbildung 26 – Skizze des Auswahlménüs	40
Abbildung 27 – Skizze einer möglichen Verkettung von Tangibles	41
Abbildung 28 – Aufgabe 1 der Vorstudie	44
Abbildung 29 – Aufgabe 2 der Vorstudie	44
Abbildung 30 – Aufgabe 3 der Vorstudie.	45
Abbildung 31 – Aufgabe 4 der Vorstudie	45
Abbildung 32 – Aufgabe 6 der Vorstudie	46
Abbildung 33 – Aufgabe 7 der Vorstudie	47
Abbildung 34 – Aufgabe 8 der Vorstudie	47
Abbildung 35 – Bild der Tabletop-Oberfläche zu Beginn.....	55
Abbildung 36 – Bild des virtuellen Menüs eines Tangibles.....	56
Abbildung 37 – Darstellung der Tangible-Verkettung auf der Tabletop-Oberfläche	57
Abbildung 38 – Skizze der Funktion eines Filter-Tangibles.....	58
Abbildung 39 – Skizze der Funktion eines Highlight-Tangibles	59
Abbildung 40 – Darstellung der Tabletop-Oberfläche.....	59
Abbildung 41 – Tangible-Prototyp „Spinne“	61
Abbildung 42 – Tangible-Prototyp „Block“ auf Steckbrett	62
Abbildung 43 – Tangible-Prototyp „Block“ mit Gehäuse	63
Abbildung 44 – Seitenansicht und Draufsicht	64
Abbildung 45 – Code des Befehlsinterpreters	65
Abbildung 46 – Algorithmus zur Schüttelgestenerkennung	66
Abbildung 47 – Struktur der Eye-Tracking-Daten.....	68
Abbildung 48 – Vereinfachtes Klassendiagramm der Architektur der Benutzeroberfläche	69
Abbildung 49 – Übersicht über alle graphischen Elemente des Prototyps	70
Abbildung 50 – Sequenzdiagramm beim Verbinden zweier Tangibles.....	71
Abbildung 51 – Codeauszug der Zuordnung von Byte Tags zu Bluetooth-Adressen.	72
Abbildung 52 – Ablauf bei der Bluetooth-Kommunikation	73
Abbildung 53 – Beispiel einer möglichen Graphstruktur aus verbundenen Tangibles	73
Abbildung 54 – Generierungsbeispiel eines Graphen.....	74
Abbildung 55 – Aufbau der Evaluationsstudie.	78
Abbildung 56 – Exemplarische Lösung der ersten Aufgabe der Evaluationsstudie	79

<i>Abbildung 57 – Exemplarische Lösung der zweiten Aufgabe der Evaluationsstudie.....</i>	<i>80</i>
<i>Abbildung 58 – Exemplarische Lösung der dritten Aufgabe der Evaluationsstudie</i>	<i>81</i>
<i>Abbildung 59 – Exemplarische Lösung der vierten Aufgabe der Evaluationsstudie</i>	<i>82</i>
<i>Abbildung 60 – Proband beim Bearbeiten einer Aufgabe während der Evaluationsstudie</i>	<i>84</i>
<i>Abbildung 61 – Szene aus dem Film Iron Man</i>	<i>91</i>

1 Einleitung

Neue technische Errungenschaften sowie Interaktionsformen können dazu führen, dass Aufgaben besser, schneller und aus Sicht des Bearbeiters mit mehr Spaß erledigt werden. Ist die Aufgabe darüber hinaus regelmäßig durchzuführen und umständlich in der Handhabung, werden Änderungs- oder Verbesserungsmaßnahmen gerne von Personen angenommen, die sich damit auseinandersetzen müssen. Eine solche Aufgabe ist das Analysieren von Eye-Tracking-Daten am Institut für Visualisierung und Interaktive Systeme der Universität Stuttgart. Mittels spezieller Soft- und Hardware [1] werden die Augenbewegungen von Probanden während der Durchführung einer Aufgabe (etwa die Auswertung einer Homepage) aufgenommen. Durch die Auswertung der Daten kann dann beispielsweise das Layout der Homepage optimiert werden.

Beim Auswerten der Daten werden einzelne Datensätze oder Gruppendatensätze in verschiedenen Visualisierungsformen wie Scan-Path-Diagrammen oder Heatmaps grafisch dargestellt, aus welchen der Analyst Rückschlüsse auf die jeweilig durchgeführte Aufgabe ziehen kann. Diese Arbeit wird entweder von jedem Analyst einzeln ausgeführt oder alle Analysten werten die Ergebnisse gemeinsam vor einem Computer aus. Heutige Computersysteme besitzen in der Regel lediglich eine Maus und eine Tastatur, sodass nicht alle Analysten die Software gleichzeitig bedienen können. Für eine kollaborative Datenuntersuchung sind diese Systeme deshalb nicht ausgelegt. *Tabletop-Computer* wie Microsoft PixelSense [2] können als Eingabegerät eines *Tangible User Interfaces* [3] dienen und somit völlig neue Interaktionsmöglichkeiten eröffnen. Mit den Tangibles können mehrere Benutzer interaktiv das System bedienen. Um die Informationen übersichtlicher darstellen zu können, kann die Ausgabe auf einem großen externen Monitor erfolgen (etwa einer Powerwall).

Diese Entwicklung vom Desktop-Computer hin zu Tangible User Interfaces deckt sich auch mit der Vorhersage von Mark Weiser im Jahre 1991: „*Specialized elements of hardware and software, connected by wires, radio waves and infrared, will be so ubiquitous that no one will notice their presence.*“ [4]. Er prophezeit eine Verschmelzung aller Computer mit der Umgebung und ein verändertes Bewusstsein der Menschen, die sich aufgrund der Allgegenwärtigkeit darüber überhaupt keine Gedanken mehr machen.

Diese Arbeit soll einen kleinen Zwischenschritt auf dem von Weiser propagierten Weg zum *ubiquitous computing* sein. Am realen Szenario einer Eye-Tracking-Datenanalyse sollen neue Interaktionsformen und –techniken implementiert werden, um die Benutzererfahrung zu verbessern. Ein Fokus wird hierbei insbesondere auf intelligente Tangibles gelegt, da in einer vorherigen Arbeit [5] bereits ein ähnlicher Ansatz gewählt wurde. Das Ziel dieser Arbeit ist es, dass durch die Tangibles ein Mehrwert für die Nutzer geschaffen wird, den er ohne Tangibles nicht hätte. Dies soll durch bidirektionale Kommunikation zwischen den Tangibles und dem Programm, welches auf dem Tabletop-Computer läuft, erreicht werden. Die Tangibles selbst werden mit Sensoren, Anzeigen und Kommunikationsgeräten ausgestattet, um dies zu erreichen (*Active Tangibles*).

1.1 Aufbau der Arbeit

Dem Leser wird nun ein kurzer Überblick gegeben, in welche Teile sich diese Arbeit gliedert und was darin beschrieben wird.

Der Einleitung folgt das Grundlagenkapitel, worin es hauptsächlich darum geht, Begriffe (unter anderem *Natural User Interfaces* und *Tangible User Interfaces*) und Techniken (unter anderem *Rapid-Prototyping-Verfahren* und *Tabletop-Implementierungstechniken*) zu

erläutern, die für das Gesamtverständnis dieser Arbeit relevant sind. Es werden verschiedene Benutzerschnittstellen- und Interaktionstechniken genannt und über deren Implementierungstechnik gesprochen. Darüber hinaus wird dies anhand zahlreicher Beispiele verdeutlicht.

Im Anschluss daran werden verwandte Arbeiten vorgestellt, die sich mit einem ähnlichen thematischen Hintergrund beschäftigen. Da das Themengebiet der Tangible User Interfaces sich in den letzten Jahren größerer Beliebtheit unter Forschern erfreut, ist es interessant zu sehen, welche Möglichkeiten bereits ausgelotet wurden. Darüber soll diese Auflistung der wichtigsten Arbeiten einen kurzen Überblick verschaffen.

Das vierte Kapitel beschreibt die Aufgabenstellung samt Hintergrundinformationen und gibt eine kurze Übersicht darüber, wie die Aufgabe gelöst worden ist.

Danach werden die im Rahmen dieser Arbeit erfassten Anforderungen vorgestellt. Dazu wurden Gespräche mit Experten geführt und eine Liste erstellt, aus der anschließend eine möglichst große Teilmenge an Anforderungen ausgewählt und im Prototyp sowie dem Lösungskonzept umgesetzt wurde. Zusätzlich dazu wurden existierende Quellen nach Anforderungen durchsucht, um die eigenen gegebenenfalls ergänzen zu können.

Der nächste Teil beschreibt den Lösungsansatz für die Aufgabenstellung. Dabei werden zuerst anfängliche Vorüberlegungen erläutert, welche in einer Vorstudie mit Probanden getestet und erweitert wurden. Für die Studie wurde ein realistisches Szenario zur Eye-Tracking-Datenanalyse erstellt, mit den Probanden durchgespielt und deren Meinungen notiert. Aus der Kombination aus Studienergebnissen und Vorüberlegungen wurde das finale Lösungskonzept erstellt, welches am Ende des Kapitels präsentiert wird.

Im siebten Kapitel wird der entwickelte Prototyp beschrieben. Dieser gliedert sich in ein Softwareprogramm, welches auf dem Tabletop-Computer läuft sowie in die Tangibles, die speziell entwickelt und programmiert wurden. Es werden Einzelheiten zur Architektur der Tabletop-Software erläutert sowie das Zusammenspiel zwischen Tabletop und Tangibles verdeutlicht.

Kapitel acht beschreibt die zweite Studie, bei der es darum geht, das entwickelte Interaktionskonzept am Beispiel von Eye-Tracking-Daten mit dem bisherigen Analysevorgang zu vergleichen.

Im den letzten beiden Kapiteln wird die Arbeit zusammengefasst und ein Ausblick auf zukünftige Verbesserungen und Weiterentwicklungen gegeben, die im Zusammenhang mit dieser Arbeit stehen.

2 Grundlagen

Für ein besseres Verständnis dieser Arbeit werden in diesem Kapitel die Grundlagen dieser Arbeit behandelt. Die nachfolgende Zeitleiste gibt einen groben Überblick über bedeutende Entwicklungen im Bereich Natural User Interfaces (im Folgenden NUI), Tangible User Interfaces und Tabletop-Computern. Einige der Beispiele werden im weiteren Verlauf dieser Arbeit ausführlicher beschrieben.

Die nächsten beiden Unterkapitel befassen sich zunächst mit Natural User Interfaces und Tangible User Interfaces. Darin wird die Interaktionstechnik der Interfaces beschrieben, auf deren besondere Eigenschaften eingegangen und diese anhand von Beispielen verdeutlicht. Im Anschluss folgt eine kurze Einführung in das Thema der Tabletop-Computer. Dabei wird der Begriff des Tabletop-Computers definiert, eine grobe Übersicht über mögliche Eingabetechniken gegeben und Beispiele gezeigt. Schließlich wird im letzten Teil dieses Kapitels zuerst kurz auf Rapid-Prototyping im Allgemeinen und danach auf relevante Prototyping-Verfahren, die in dieser Arbeit verwendet wurden, eingegangen.

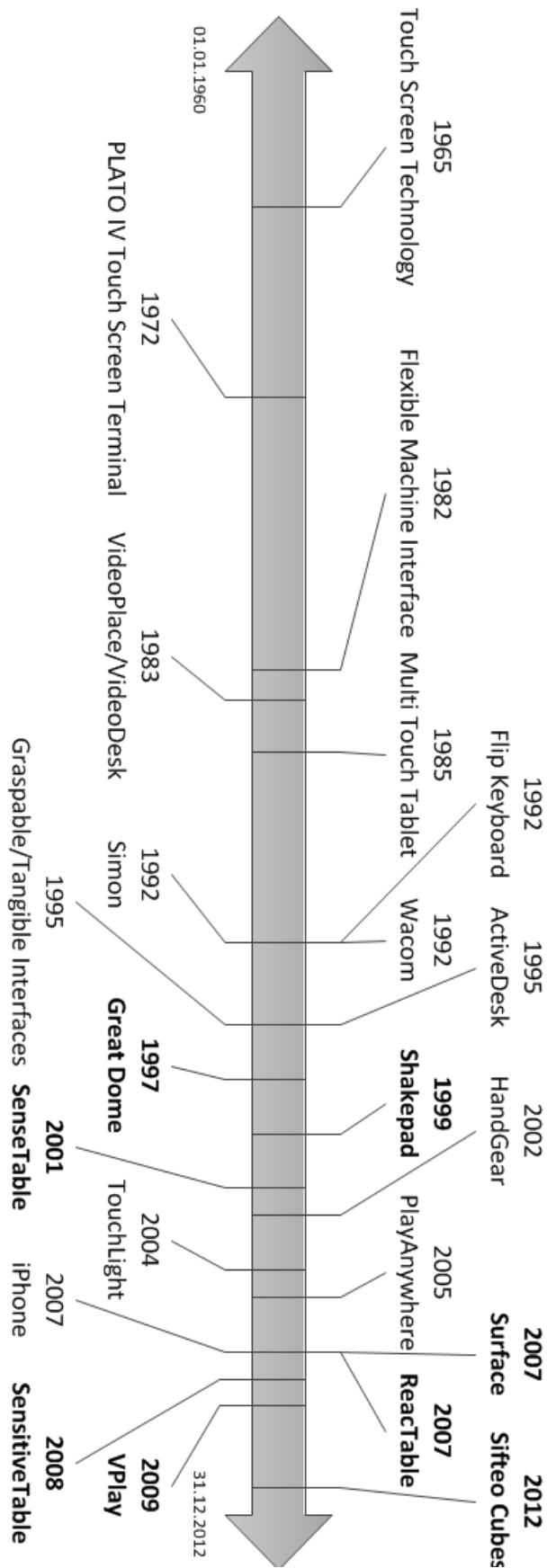


Abbildung 1 – Die Grafik zeigt eine zeitliche Abfolge von wichtigen Entwicklungen im Bereich Natural User Interfaces, Tangible User Interfaces und Tabletop-Computern von Anfang 1960 bis Ende 2012. Fett gedruckte Einträge werden im weiteren Verlauf explizit erwähnt. Quelle: [6]

2.1 Natural User Interfaces

Im Jahr 1999 beschrieb Westerman [7] die grundlegenden Interaktionsformen von Natural User Interfaces, ohne jedoch den Begriff explizit zu erwähnen. Wichtige Arbeiten zum NUI-Paradigma entstanden allerdings teilweise über 10 Jahre früher [6]. Natural User Interfaces verändern die Interaktionsart in eine Richtung, die für Menschen natürlicher erscheinen soll. Dafür werden Finger- und Gesten-Interaktionen verwendet. Anstatt eines Zeigegeräts benutzt er zum Selektieren eines Dokuments seine Finger. Dieses Prinzip wird *direkte Manipulation* genannt und ist ein zentrales Paradigma bei Natural User Interfaces. Besonders für Anfänger eignet sich diese Interaktionsform gut, da sie intuitiver und leichter erlernbar im Vergleich zum WIMP-Modell (*Windows, Icons, Menues, Pointer*) erscheint. [8] Heutige grafische Benutzerschnittstellen werden von Natural User Interfaces nicht komplett ersetzt werden können, da beide Typen ihre eigenen Stärken und Schwächen haben. So ist es beispielsweise schwierig, eine Grafik mittels Gesten pixelgenau an einer Linie auszurichten, wohingegen die Interaktion mit Hilfe einer Maus dazu in der Lage ist. Entsprechend ist es aber unkomplizierter, ein öffentliches Terminal als Natural User Interface mit Finger- und Gesten-Interaktion zu konzipieren und auf traditionelle Eingabegeräte wie Maus und Tastatur zu verzichten. Ähnlich wie beim Übergang von *Command Line Interfaces (CLI)* zum grafischen Interface werden beide System koexistieren und entsprechend ihren Stärken eingesetzt werden. [9] Populäre Beispiele für Natural User Interfaces sind Terminals an öffentlichen Orten wie zum Beispiel Bank- oder Fahrkartenautomaten sowie Smartphones und Tablet Computer, die per Finger- oder Gestensteuerung bedienbar sind.

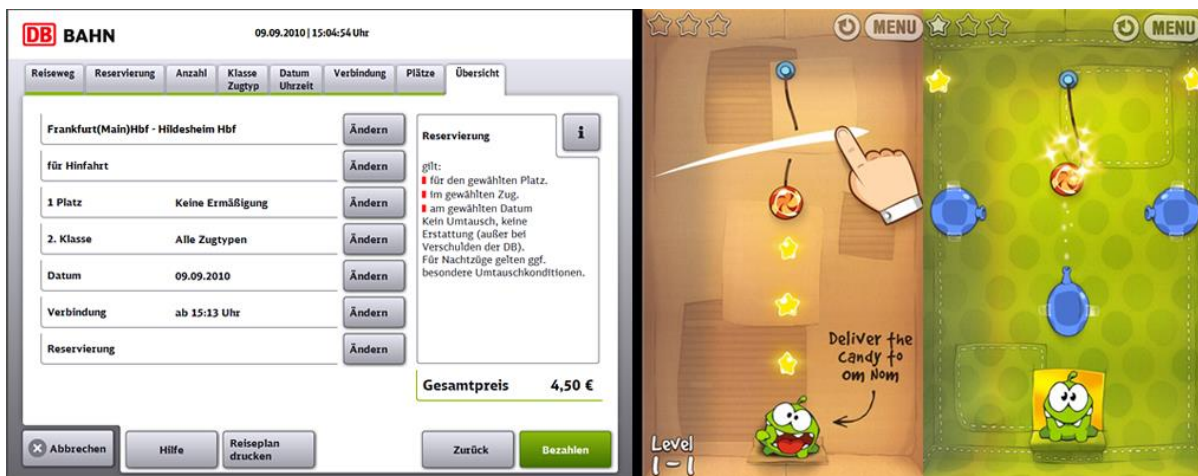


Abbildung 2 – Beispiele für Natural User Interfaces. Im linken Bild ist die Benutzeroberfläche eines Fahrkartenautomaten der Deutschen Bahn zu sehen, im rechten Bild das populäre Smartphone-Spiel *Cut the Rope*. Quellen: Deutsche Bahn AG¹, WMPowerUser²

¹ <http://www.bahn.de/p/view/service/vertriebswege/automat/ntaw.shtml>

² <http://wmpoweruser.com/wp-content/uploads/2012/11/Cut-The-Rope-Windows-Phone.jpg>

2.2 Tangible User Interfaces

Der Begriff Tangible User Interface wurde zum ersten Mal in einer Arbeit von Hiroshi Ishii und Brygg Ullmer [3] erwähnt. Darin beschreiben sie eine Benutzerschnittstelle, die aus den Objekten der realen Welt besteht und mit digitalen Informationen angereichert wird. Das Ziel dieser Art von Benutzerschnittstelle ist die Kopplung zwischen den Daten der virtuellen und den Objekten der realen Welt. Ihre Vision war es, jegliche Oberfläche (Wände, Decken, Türen, Tische, ...) der realen Welt mit der virtuellen zu verbinden. Physische Objekte wie Karten, Bücher oder Gläser sollten mit digitalen Informationen angereichert werden und die Interaktion mit selbigen sollte Rückkopplungen bewirken. Somit wollten sie ein Verwischen des Übergangs zwischen virtueller und realer Welt erreichen.

Ein Tangible User Interface ist eine Benutzerschnittstelle, die es erlaubt, kollaborativ mit physischen Objekten zu interagieren, um digitale Daten zu verändern. Die Benutzeroberfläche wird somit greifbar (*tangible, graspable*). Da prinzipiell jedes Objekt als Tangible benutzt werden kann, können auch mehrere Personen gleichzeitig ein System mit diesem Interface nutzen. Die treffendste Übersetzung für Tangible auf Deutsch wäre „haptisches Objekt“. Einige Beispiele zu Tangible User Interfaces verwenden zur Umsetzung einen Tabletop-Computer, der die virtuelle Welt repräsentiert und deren Inhalte darstellt (siehe Kapitel 3). Im Folgenden werden drei Beispiele für Tangible User Interfaces vorgestellt.

Great Dome

Der Great Dome [10] ist ein System, welches das Zusammenspiel von Tabletop-Computern (siehe Kapitel 2.3) und Tangibles verdeutlicht. Ein Miniaturmodell des Great Domes (Campusgebäude des Massachusetts Institute of Technology) wird auf der Oberfläche des Computers platziert. Daraufhin erscheint eine zweidimensionale Karte, die so ausgerichtet ist, dass die Position des Tangibles mit dem Standort des Great Dome auf der Karte übereinstimmt. Der Nutzer kann den Great Dome verschieben und die Karte richtet sich dementsprechend neu aus.



Abbildung 3 – Great Dome (MIT). Der Benutzer kann mithilfe eines Modells des Great Domes als Tangible die Ausrichtung einer 2D Karte des MIT Campus steuern. Die Karte richtet sich immer so aus, dass die Position des Tangibles mit der des Great Domes auf der Karte übereinstimmt. Quelle: [10]

Shakepad

Das Shakepad [10] besteht aus einer 8x8 LED Displaymatrix, einem Mikroprozessor und Beschleunigungssensoren. Die Bewegungen, die der Nutzer mit dem Pad in der Hand ausführt, werden registriert und in eine grafische Ausgabe auf dem Display umgewandelt. Neigt der Nutzer das Gerät in eine Richtung, werden die Pixel in dieser Richtung farbig dargestellt (ähnlich einem *Snake*-Spiel auf älteren Mobiltelefonen). Um die Eingabe zu löschen, genügt ein Schütteln des Pads, daher auch dessen Name.



Abbildung 4 – Bewegungen werden in eine grafische Ausgabe auf dem Shakepad umgesetzt (ähnlich einem Snake-Spiel auf älteren Mobiltelefonen), ein Schütteln löscht die Anzeige. Quelle: [10]

Sifteo Cubes

Sifteo Cubes [11] sind kleine, etwa 4 Zentimeter große Würfel mit einem Bildschirm auf einer Seite. Ein Würfel registriert, wenn er in der Nähe eines anderen Würfels ist, sodass etwa die Anzeige geändert werden kann. Ein externer Computer kann die Würfel über WLAN ansprechen und einrichten. Auf diese Weise können Spiel- und Lernanwendungen implementiert und mittels der Würfel umgesetzt werden.



Abbildung 5 – Sifteo Cubes reagieren auf Annäherung zueinander und passen ihre Anzeige an. Quelle: [11]

2.3 Tabletops

Ein Tabletop-Computer ist ein Computer, der in einem Tisch verbaut wird. Die Oberfläche dient als Bildschirm und ist je nach Eingabetechnik (siehe 2.3.1) aus einem entsprechenden Material gefertigt. Darüber hinaus beinhaltet sie eine Schicht, die Projektionen von unten oder oben abbilden kann, um eine Ausgabe von etwa Daten, Bildern, Tabellen und Texten zu ermöglichen. Eingaben erfolgen typischerweise über Berührung, Gesten und Tangibles. Je nach verwendetem Verfahren können Objekte aber auch im Raum über dem Tisch erkannt sowie über Tags identifiziert und verfolgt werden.

2.3.1 Eingabetechniken und Funktionsweise

Multi-Touch-Oberflächen können auf mehrere Arten implementiert werden. Schöning et al. [12] gehen in ihrem Artikel genauer darauf ein und stellen zahlreiche Techniken mit ihren Vor- und Nachteilen vor. Dieses Kapitel fasst die Ergebnisse aus ihrer Arbeit kurz zusammen.

Resistive Technik

Ein resistiver Touchscreen arbeitet mit zwei elektrisch leitenden Schichten, die durch Abstandshalter getrennt sind. Die äußere Schicht ist flexibel und lässt sich per Berührung biegen. Die innere Schicht besteht dagegen aus einem harten glasähnlichen Material. Dazwischen befindet sich eine Isolierschicht aus Silikonpunkten. Drückt der Nutzer nun auf einen Punkt auf der Oberfläche, werden die beiden leitenden Schichten verbunden und eine elektrische Spannung wird registriert. Eine Kontrolleinheit kann nun daraus den genauen Punkt der Berührung ermitteln.

Die Genauigkeit von resistiven Touchscreens ist im Vergleich zu kapazitiven Systemen geringer und die flexible äußere Schicht ist anfällig für Beschädigungen. Allerdings benötigt das Verfahren wenig Energie zum Betrieb. Digitale Kameras und Personal Digital Assistants (PDA) setzen es häufig ein.

Kapazitative Technik

Kapazitative Touchscreens sind im Gegensatz zu resistiven Touchscreens teurer in der Produktion, allerdings auch widerstandsfähiger und genauer. Sie werden daher in der Industrie oder als öffentliche Bildschirme eingesetzt. Zur Bedienung dieser Art von Bildschirmen wird ein Gegenstand mit kapazitiven Eigenschaften benötigt, um eine Spannungsänderung hervorzurufen. Es gibt zwei Kategorien von kapazitiven Verfahren: das Verfahren mit *projizierender Kapazität* und das *Oberflächenkapazitätsverfahren*.

Bei der *projizierenden Kapazität* werden in einer Zwischenschicht dünne elektrisch leitende Kabel als Matrix verlegt. Eine Richtung dient als Signalsender, die orthogonale Richtung dazu als Empfänger. Diese Zwischenschicht kann von nichtleitendem Material ummantelt werden, was die Haltbarkeit erhöht. Durch eine Berührung kann die genaue Position aus der Spannungsänderung heraus berechnet werden. Die Genauigkeit der beiden Verfahren ist sehr ähnlich.

Beim *Oberflächenkapazitätsverfahren* wird eine geringe elektrische Spannung auf einer leitenden durchsichtigen Schicht erzeugt. Diese Spannung geht von den vier Eckpunkten der Fläche aus. Wird die Fläche nun mit einem kapazitiven Gegenstand (etwa dem menschlichen Finger) berührt, wird eine Spannungsänderung bei den vier Sensoren in den Ecken registriert. Da jeder Sensor einen anderen Wert erhält, kann ein Mikroprozessor daraus die genaue Position des Druckpunktes berechnen.

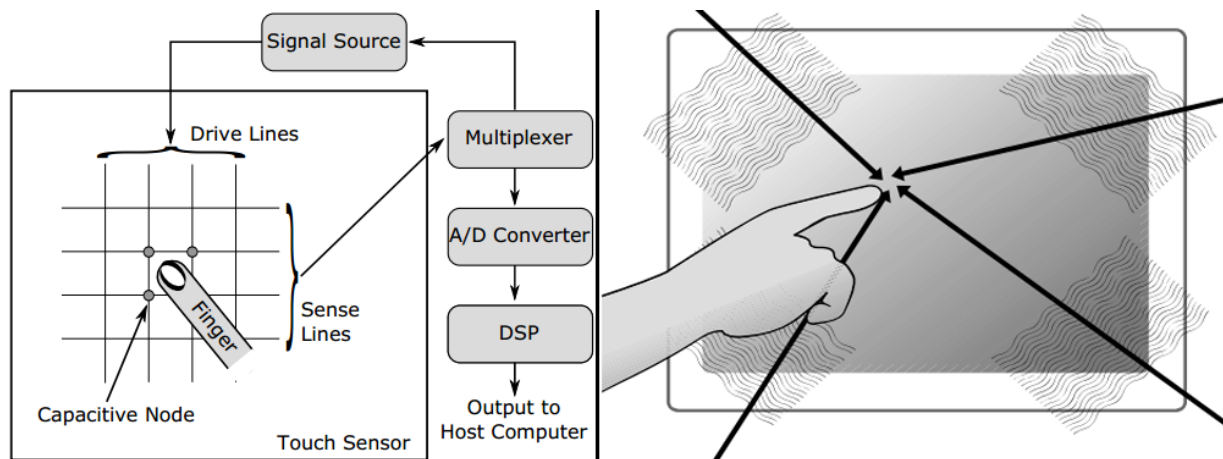


Abbildung 6 – Kapazitive Techniken im Vergleich. Im linken Bild die Umsetzung beim Verfahren mit projizierender Kapazität, im rechten Bild ist die Ermittlung des Berührungspunktes beim Oberflächenkapazitätsverfahren zu sehen. Quelle: [12]

Frustrated Total Internal Reflection-Technik (FTIR)

Frustrated Total Internal Reflection basiert darauf, dass elektromagnetische Wellen innerhalb einer Schicht reflektiert werden. Dazu muss die umgebende Schicht einen geringeren Brechungsindex als das Material selbst haben und der Auftreffwinkel der Wellen muss genügend gering sein. Ein typischer Aufbau besteht aus einer glasähnlichen Schicht mit Infrarotseindern an den Seiten. Das ausgesendete Licht wird an den Rändern komplett reflektiert und verbleibt innerhalb der Glasschicht. Berührt nun der Nutzer mit einem Finger die Oberfläche, wird an dieser Stelle der Brechungsindex verändert und das Licht tritt aus der Glasschicht heraus. Eine Kamera unterhalb der Oberfläche fängt das reflektierte Licht auf und kann somit die Stelle ermitteln, an welcher eine Berührung stattgefunden hat. Der Projektor ist für die Darstellung zuständig. Han et al. [13] machte dieses Verfahren 2005 im Zusammenhang mit Multi-Touch populär.

Das Verfahren ist sehr zuverlässig und robust, allerdings können trockene Haut sowie Ablagerungen (Staub, Schmutz) auf der Oberfläche durch lange Nutzungsdauer die Ergebnisse beeinträchtigen. Da das System mit Infrarotlicht arbeitet, sind zusätzliche Infrarotquellen in der Umgebung für die Ergebnisqualität ebenfalls hinderlich.

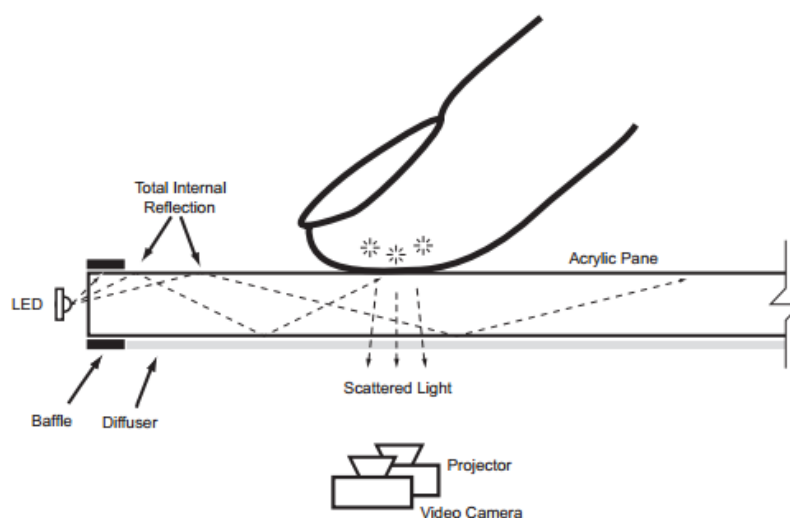


Abbildung 7 – Schematische Darstellung der Frustrated Total Internal Reflection-Technik (FTIR). Die Berührung durch den Finger ändert die Brechung an der Stelle und das aus der Glasschicht ausströmende Infrarotlicht wird von der Kamera erfasst. Durch die Erfassung kann die Position des Berührungspunktes ermittelt werden. Quelle: [13]

Diffuse Illumination-Technik (DI)

Im Gegensatz zu FTIR wird beim *Diffuse Illumination*-Verfahren die Tabletop-Oberfläche von hinten mit Infrarotlicht bestrahlt. Eine Kamera fängt das von Objekten reflektierte Licht auf, die mit der Oberfläche Kontakt haben. Je nach Stärke des ausgesendeten Lichts und der Durchlässigkeit der Projektionsfläche können auch Objekte hinter der Fläche (bei einer horizontalen Tischpositionierung also oberhalb der Tischoberfläche) erfasst werden. Typischerweise wird diese Fläche mit robustem Glas und einer transparenten Sicherheitsschicht versehen, um Beschädigungen zu vermeiden. Um Objekte eindeutig identifizieren zu können, kann auf deren Unterseite ein Identifizierungsmarker angebracht werden, der dies gewährleistet.

Das Verfahren ist robust, leidet aber wie FTIR darunter, wenn in der Umgebung Infrarotstrahlung vorhanden ist. Je gleichmäßiger die Oberfläche mit Infrarotlicht bestrahlt wird, desto besser sind die Ergebnisse.

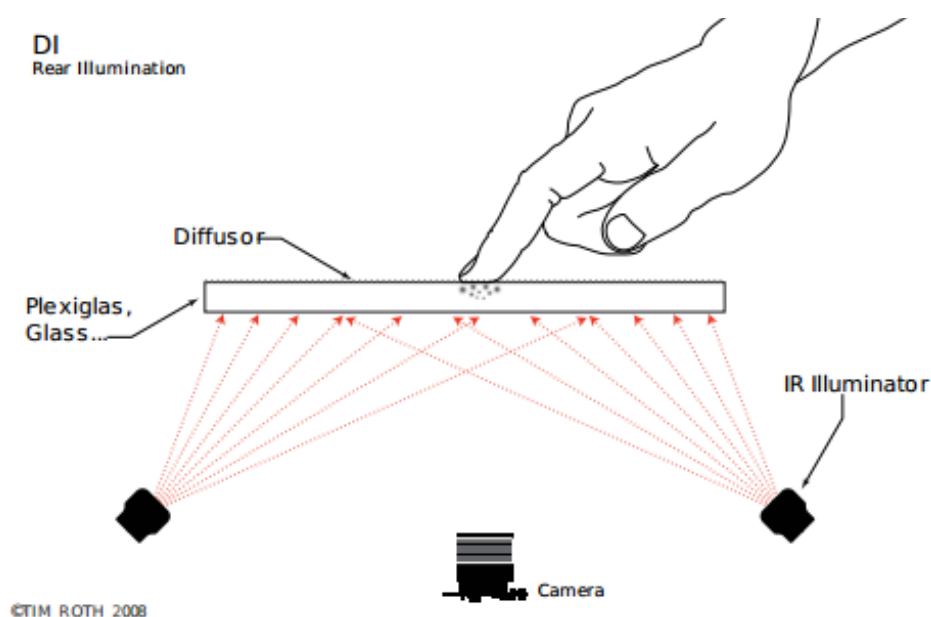


Abbildung 8 – Schematische Darstellung der Diffuse Illumination-Technik (DI). Die Oberfläche wird von hinten mit Infrarotlicht bestrahlt und eine Kamera fängt Reflexionen durch Objekte auf, die sich auf oder vor der Oberfläche befinden. Quelle: [12]

Diffused Surface Illumination-Technik (DSI)

Diffuse Surface Illumination stellt eine weitere optisch basierte Erkennungstechnik dar und funktioniert ähnlich wie die Diffuse Illumination-Technik. Dabei wird eine spezielle Glasschicht verwendet, in welcher kleine Teilchen enthalten sind. Diese Teilchen arbeiten als eine Art Spiegel und verteilen das Infrarotlicht gleichmäßig innerhalb der Glasschicht. Das Licht wird hierbei wie beim FTIR-Verfahren seitlich in die Schicht eingespeist.

Jedes FTIR-System lässt sich leicht in ein DSI-System umbauen, um eine gleichmäßige Lichtverteilung zu erhalten. Allerdings ist dieses Verfahren anfälliger für Infrarotlicht in der Umgebung als andere Verfahren und der Kontrast ist geringer als beim DI-Verfahren, da Infrarotlicht zur Kamera reflektiert wird.

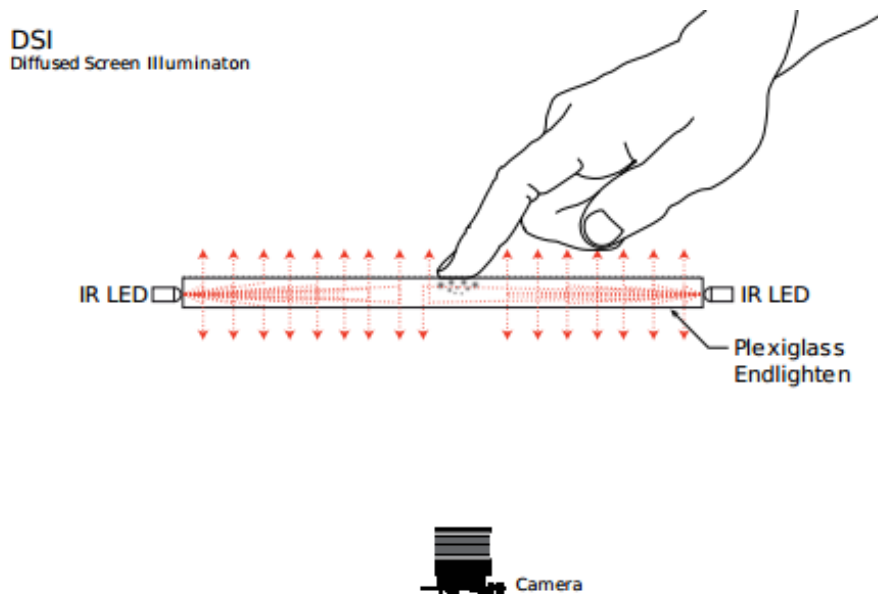


Abbildung 9 – Schematischer Aufbau der Diffused Surface Illumination-Technik (DSI). Eine spezielle Glasschicht wird seitlich mit Infrarotlicht durchflutet und kleine Teilchen innerhalb agieren als Spiegel, um das Licht gleichmäßig zu verteilen. Quelle: [12]

Microsoft PixelSense-Technik

Diese optische Technik benötigt im Gegensatz zu den vorherigen Verfahren keine separate Kamera. Das System arbeitet mit Infrarotlicht, welches von unten in Richtung der Oberfläche gestrahlt wird. Wird nun ein Gegenstand auf die Oberfläche gestellt oder in geringem Abstand darüber gehalten, erfassen integrierte Sensoren unterhalb des Sicherheitsglases das reflektierte Licht. Die Daten aller Sensoren zusammen ergeben das Sensorabbild, das ein Prozessor nun entsprechend verarbeiten kann.

Dieses Verfahren ist ebenfalls anfällig für Umgebungsinfrarotlicht, eignet sich aufgrund des Aufbaus allerdings auch für platzsparende Tabletop-Computer. [2]

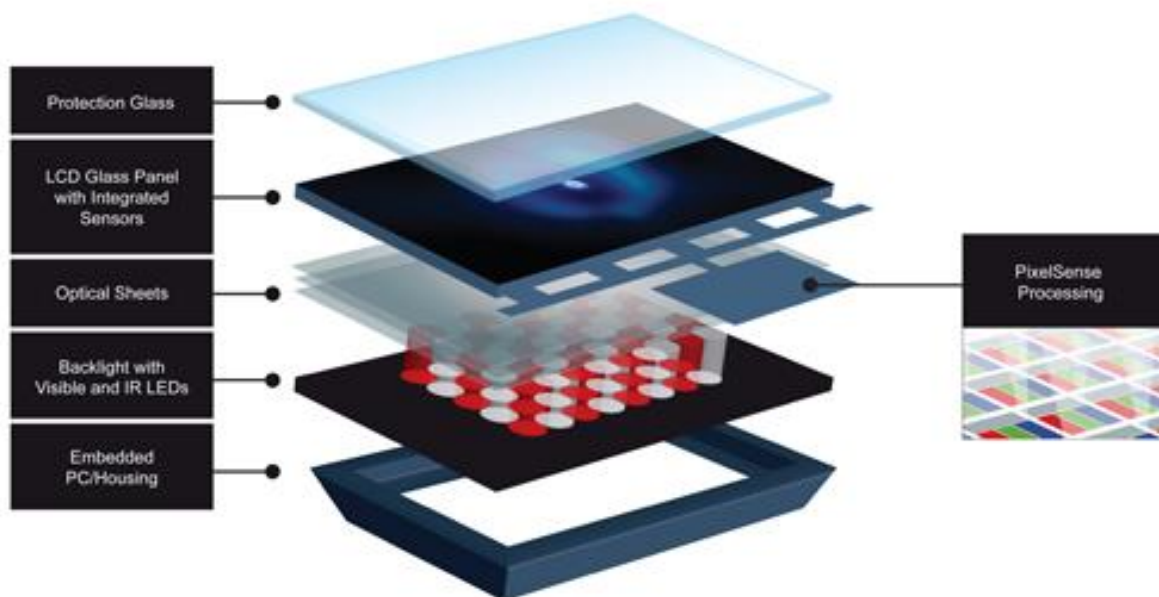


Abbildung 10 – Schematischer Aufbau der PixelSense-Technik. Infrarotlicht wird in Richtung der Oberfläche gestrahlt und integrierte Sensoren erfassen die Reflexionen von Objekten. Quelle: [2]

2.3.2 Beispiele

In diesem Kapitel werden einige Beispiele für Tabletop-Computer vorgestellt und beschrieben.

Microsoft PixelSense

PixelSense (frühere Bezeichnung Surface) ist ein Tabletop-Computer der Firma Microsoft. Die erste Version wurde 2007 veröffentlicht, die aktuelle Version 2011. Der aktuelle Tisch wird von Samsung unter dem Codenamen SUR40 gefertigt, besitzt eine Dicke von etwa 10 Zentimetern und nutzt die PixelSense-Eingabetechnik von Microsoft (siehe Kapitel 2.3.1). Als Betriebssystem kommt Windows 7 zum Einsatz, jedoch stehen dem Tisch zusätzliche Demo- und Analyseprogramme zur Seite, die für Finger- und Gesten-Interaktion optimiert sind. Zum Entwickeln von Anwendungen steht das Surface SDK in der Version 2.0 bereit. Um Alltagsgegenstände als Tangibles auf dem PixelSense zu nutzen, können diese mit einem Byte Tag [14] versehen werden. Wenn der Tag auf der Tischoberfläche platziert wird, kann das System das Objekt erkennen und verfolgen.

Die geringe Dicke des Tisches ermöglicht sowohl die Verwendung in horizontaler Position mit Standfüßen als auch in vertikaler Position an einer Wand. Er wurde unter anderem von Cafés, Banken und Hotels im praktischen Einsatz genutzt. [15]



Abbildung 11 – Tabletop-Computer PixelSense der ersten Generation von Microsoft im Einsatz. Benutzer können Objekte mit Tags versehen und sie auf die Oberfläche legen. Das System zeigt dazu passende Zusatzinformationen an. Quelle: [Wikimedia.org](http://upload.wikimedia.org/wikipedia/commons/8/88/Surface_table.JPG)³

SensitiveTable

Der SensitiveTable besitzt eine etwa 140 Zentimeter große Anzeige mit optischer Multi-Touch-Oberfläche. Im Inneren des Gehäuses sitzt ein Standard-Computer, der Rechenleistung für den Betrieb des Systems zur Verfügung stellt. Weitere Elemente sind Mikrofone zur Spracherkennung und RFID-Antennen zur Interaktion mit Objekten mit RFID-Sendern. Der Tisch ist für Media Browsing (etwa die Interaktion mit vielen Bildern) optimiert und kann an öffentlichen Plätzen kollaborativ verwendet werden. [16]

³ http://upload.wikimedia.org/wikipedia/commons/8/88/Surface_table.JPG



Abbildung 12 – Natural Interaction SensitiveTable als MediaBrowser. Ein Nutzer arbeitet mit mehreren Bildern, die durch Gesten manipuliert werden können. Quelle: [16]

2.4 Techniken zur Entwicklung von Tangibles

Für die Entwicklung von Prototypen gibt es mehrere Standards in der Industrie. In dieser Arbeit sollen Tangibles erstellt werden, wobei auf diverse Standards zurückgegriffen wird. Zuerst folgt eine kurze Einführung in Rapid-Prototyping. Anschließend werden Verfahren vorgestellt, die in dieser Arbeit verwendet wurden.

2.4.1 Einführung in Rapid-Prototyping

Rapid-Prototyping zielt darauf ab, möglichst einfach und schnell ein kostengünstiges reales Produkt eines virtuellen Modells zu erstellen. Dafür wird das Modell in dünne Schichten unterteilt (Slicing) und ein Prototyper erstellt das Endprodukt. Aufgrund des Schichtenaufbaus werden die Verfahren auch *generative Verfahren* genannt. Das Verbinden der einzelnen Schichten erfolgt meist durch Kleben oder Verschmelzen. Je nach Anwendungsfall ist ein anderes Verfahren von Vorteil. Eine gute Übersicht über Rapid-Prototyping-Verfahren ist von Gebhardt et al. verfasst worden [17]. Die Übersicht zeigt die Funktionsweise sowie Vor- und Nachteile der einzelnen Techniken. Das folgende Kapitel gibt einen Überblick über die Verfahren, die in dieser Arbeit verwendet wurden, um Tangibles zu entwickeln.

2.4.2 Rapid-Prototyping-Verfahren

Es folgt ein Überblick über Verfahren, die im Rahmen dieser Arbeit eingesetzt wurden.

3D-Druck

Beim 3D-Druck werden dreidimensionale Erzeugnisse von einem 3D-Drucker produziert. Als Materialien kommen Kunstharze, Kunststoffe, Keramiken und Metalle zum Einsatz.

Das zu produzierende Objekt wird mittels 3D-Modellierungssoftware erstellt und anschließend vom Drucker produziert. Dabei können mehrere Drucktechnologien verwendet werden, die sich in Kosten, Druckdauer, Genauigkeit und einsetzbaren Materialien unterscheiden. Eine für Privatanwender interessante Technik ist das Fused Deposition Modeling (FDM). Dabei wird das Rohmaterial in der Druckerdüse geschmolzen und auf die

Bauplattform schichtweise aufgetragen. Durch das Abkühlen erhält das Produkt seine Festigkeit.

Mit diesem Verfahren ist es sehr einfach, Objekte zu erstellen. Das reicht von Schalen für Mobiltelefone⁴ über Waffenteile⁵ bis hin zu einer Herzpumpe⁶. Das Verfahren kann außerdem für Hüllen, Gehäuse und Ummantelungen genutzt werden, erlaubt jedoch nicht die Einbindung von Programmierlogik in das Element. [18]

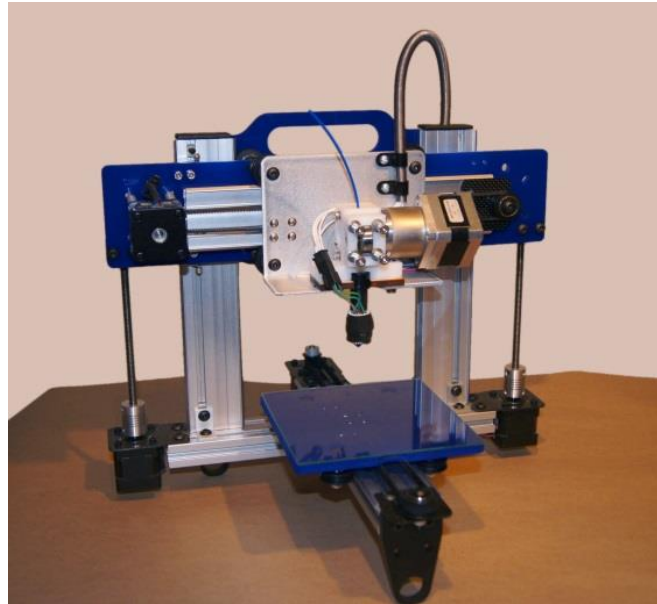


Abbildung 13 – Bild eines 3D-Druckers. Das Material wird beim Fused Deposition Modeling-Verfahren von einer Zuleitung zur Düse transportiert, verflüssigt und schichtweise auf der Ablage aufgetragen, bis das fertige Objekt entsteht. Quelle: Wikimedia.org⁷

Arduino

Das Arduino System ist ein Open Source Konzept zum einfachen Entwickeln von interaktiven Objekten für Bastler, Künstler, Designer und Interessierte. Das zentrale Mainboard beinhaltet einen Mikroprozessor, besitzt eine offene Hardwarespezifikation und verfügt über analoge und digitale Ein- und Ausgänge zum Anschließen von Modulen (*Shields*), womit es erweitert werden kann. Zum Verbinden von Modulen können Kabel eingesetzt werden, die in der Regel durch Löten an den Platinen fixiert werden. Für die Entwicklung von arduino-basierten Systemen werden elektrotechnische Kenntnisse vorausgesetzt.

Der Programmcode wird mit der Programmiersprache Processing geschrieben und per USB auf das Mainboard transferiert. Eine `setup()`-Methode wird einmalig zu Beginn ausgeführt und eine `loop()`-Methode wird permanent durchlaufen, solange das Mainboard mit Strom versorgt ist. So können beispielsweise in der Loop-Methode Leuchtdioden angesteuert werden, um deren Helligkeit permanent zu ändern.

Die Vorteile des Systems gegenüber vergleichbaren Ansätzen sind die niedrigen Anschaffungskosten, die Plattformunabhängigkeit, die einfache Erweiterbarkeit durch Module sowie die offene Architektur. Das System erlaubt die Entwicklung programmiertechnisch komplexer Objekte. [19]

⁴ <http://www.zdnet.de/88140287/nokia-liefert-templates-fur-3d-druck-von-lumia-820-hüllen/>

⁵ <http://www.golem.de/news/3d-druck-toedliche-schusswaffe-zum-selberdrucken-1208-94072.html>

⁶ <http://www.devicemed.de/fertigungseinrichtung-produktionstechnik/articles/386789/>

⁷ http://upload.wikimedia.org/wikipedia/commons/2/27/ORDbot_quantum.jpg



Abbildung 14 – Arduino Mainboard mit digitalen Ein- und Ausgängen, einem USB-Anschluss, einem 5V-Stromanschluss und einem ATmega328 Mikrocontroller. Quelle: Wikimedia.org⁸

.NET Gadgeteer Kit

.NET Gadgeteer Kit ist eine Rapid-Prototyping-Plattform, um elektronische Geräte schnell, einfach und kostengünstig herzustellen. Die Hardware ist modular aufgebaut, die Programmierung erfolgt über High-Level-Sprachen und eine Unterstützung für 3D-Design-Programme ist ebenfalls enthalten.

Der Entwickler kann aus einer breiten Palette an Modulen auswählen, die über Sockets an das Mainboard gekoppelt werden. Jedes Socket besitzt eine Nummer und ist zu Sockets mit der entsprechenden Nummer kompatibel. Man kann auch ohne elektrotechnische Grundkenntnisse bedenkenlos mit dieser Plattform entwickeln. Zur Programmierung steht ein leistungsfähiges Software-Development-Kit zur Verfügung, mit der ein eventbasiertes System entwickelt werden kann. Die erstellte Software wird per USB auf das Mainboard übertragen und das Gerät ist sofort einsatzbereit.

.NET Gadgeteer Kit ist flexibel, vielseitig einsetzbar und erlaubt einen schnellen Einstieg. Für sehr kleine Formfaktoren kommt das System aber nicht in Frage, da das Mainboard eine gewisse Mindestgröße besitzt. Ähnlich wie die Arduino Plattform lassen sich auch mit dem .NET Gadgeteer Kit programmiertechnisch komplexe Objekte erstellen. [20]

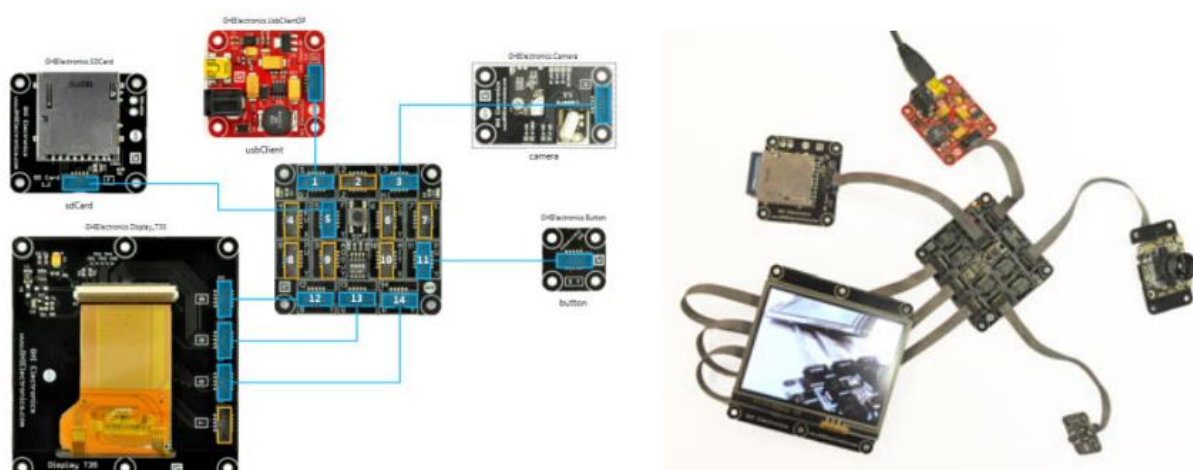


Abbildung 15 – Beispiel einer Digitalkamera mit dem .NET Gadgeteer Kit. Links die Darstellung der Module in der Entwicklungsumgebung, rechts die verbundenen Module auf einem Mainboard. Quelle: [20]

⁸ <http://upload.wikimedia.org/wikipedia/commons/6/60/ArduinoUno.jpg>

3 Verwandte Arbeiten

In diesem Kapitel werden Arbeiten vorgestellt, die mit dieser Arbeit verwandt sind. Dabei handelt es sich um Projekte, in denen mittels haptischer Objekte auf einem Tabletop-Computer kollaborativ Daten visualisiert und analysiert werden können.

VPlay

Ein Video-Jockey (VJ) ist ein Künstler, der, ähnlich dem Disc-Jockey (DJ) bei Musik, live und in Echtzeit Videos mit Effekten verbindet, um vor allem in Diskotheken und Nachtclubs für optische Effekte zu sorgen. Typischerweise arbeiten die Künstler mit Laptop und Software, um diese Videos zu produzieren. Taylor et al. [21] entwickelten das System *VPlay*, das es erlaubt, die Aufgabe auf einem Tabletop-Computer durchzuführen. Auf dem Tisch können verschiedene Objekttypen platziert werden, um den Videodatenfluss zu steuern. Bei Annäherung zweier Objekte verbinden sich diese gemäß einem Regelwerk und erzeugen dabei eine Kombination von Effekten. Ein Ausgabefenster zeigt in Echtzeit an, was der VJ produziert und wird typischerweise auf eine große Wandfläche projiziert.

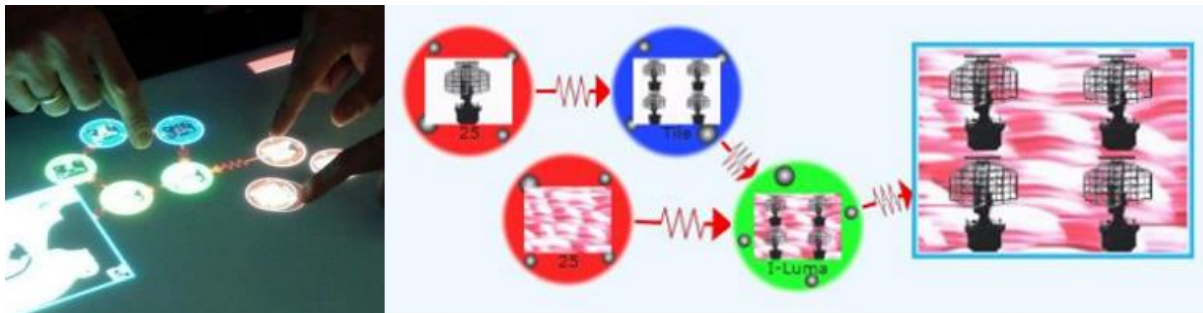


Abbildung 16 – Das linke Bild zeigt die Benutzeroberfläche von *VPlay*. Zwei Benutzer interagieren gleichzeitig mit der Oberfläche. Im rechten Bild ist die Verkettung von verschiedenen Objekttypen dargestellt, die die Ausgabe steuern. Quelle: [21]

VPlay erlaubt kollaboratives Arbeiten und setzt in der ersten Version auf ein rein grafisches Interface. Zur Gestenerkennung wird das FTIR-Verfahren (siehe Kapitel 2.3) eingesetzt. Eine Kombination mit Tangibles ist für zukünftige Versionen angedacht. Exemplarisch wird dabei eine Art Drehknopf beschrieben, der als *Scratcher* benutzt werden kann.



Abbildung 17 – Ein Beispiel für ein Tangible, das mit *VPlay* in Zukunft verwendet zur dynamischen Verzerrung von optischen Effekten verwendet werden könnte. Quelle: [21]

ReactTable

Der ReacTable ist ein runder Tisch, auf dem Nutzer kollaborativ Musik produzieren können, indem physische Objekte gesetzt, gedreht und verschoben werden. Objekte verbinden sich gemäß verschiedenen Regeln, wenn sie in der Nähe zueinander platziert werden, um eine Variation in der Audioausgabe zu erzielen. Jedes Objekt repräsentiert einen digitalen Synthesizer zur Klangerzeugung sowie einen visuellen als optisches Feedback auf der Tabletop-Oberfläche. Es stehen sechs verschiedene Synthesizer-Typen bereit, um den Klang zu manipulieren. Jeder Typ besitzt eine unterschiedliche haptische Form sowie ein unterschiedliches Symbol auf der Oberfläche. Der Disc-Jockey weiß somit immer, mit welchem Synthesizer-Typ er gerade arbeitet.

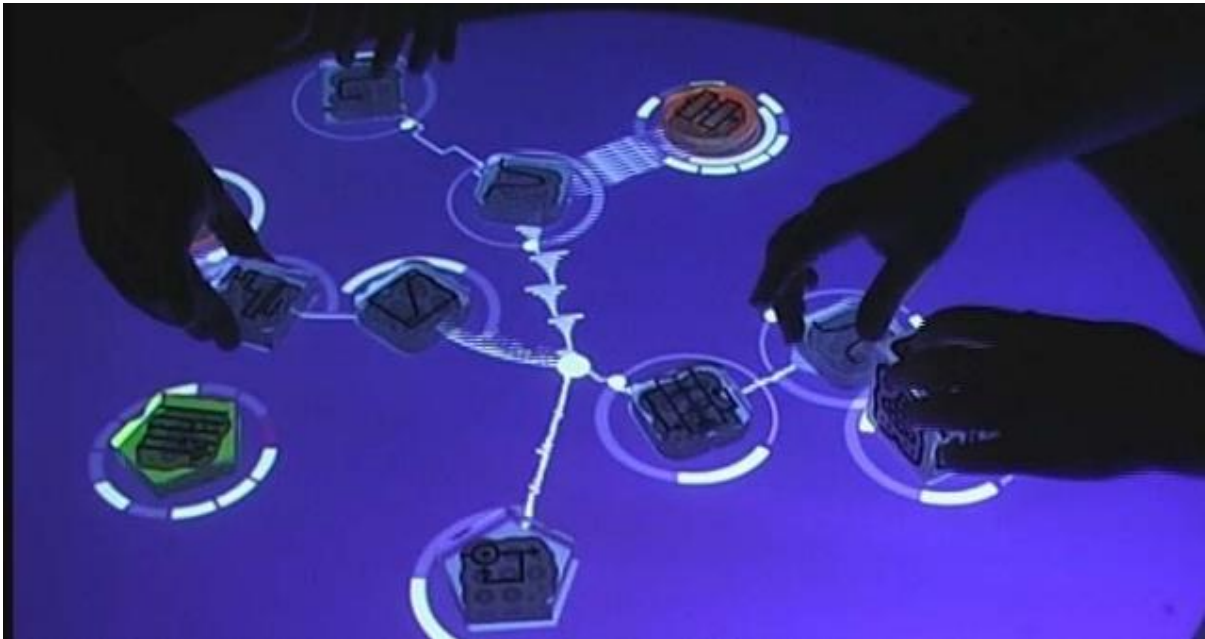


Abbildung 18 – Kollaboratives Arbeiten auf dem ReacTable. Zwei Benutzer verändern gleichzeitig die Audioausgabe durch die Interaktion mit den Synthesizer-Tangibles. Quelle: [22]

Eine Kamera unterhalb der Tischoberfläche erkennt Marker, die an der Unterseite der Objekte angebracht sind. Anhand dieser Marker kann das System Verbindungen zwischen benachbarten Objekten erstellen und darstellen.



Abbildung 19 – Vier verschiedene Marker-Typen stehen für Objekte zur Verfügung. Diese sind an der Unterseite angebracht, sodass eine Kamera die Marker erfassen und das System Verbindungen zwischen kompatiblen Objekten erstellen kann. Quelle: [22]

Mit einem Generatorelement können Klänge erzeugt und mit Mixern und Controllern verändert werden. Die Daten fließen vom Generator zum zentralen Ausgabeelement, der als weißer Punkt in der Mitte des Tisches realisiert ist. Zusätzliche Einstellungen können über Kontrollpunkte, die im Kreis um ein Objekt platziert sind, vorgenommen werden. Der Nutzer erfasst diesen Punkt per Berührung und verschiebt ihn entlang des Kreises. Damit kann beispielsweise die Abspielgeschwindigkeit oder Tonfrequenz geändert werden. [22]

SenseTable

2001 stellten Patten et al. den *SenseTable* vor [23]. Er besteht aus zwei hochauflösenden Tablet-Computern zur Erkennung von Objekten. Zusätzlich beinhaltet das System noch zwei Computer, zwei Bildschirme an der Tischhinterseite und einen Projektor direkt über dem Tisch. Ein Computer wird zur Objektverfolgung und zur Darstellung von Visualisierungen auf der Tabletop-Oberfläche benutzt. Der andere sorgt für die Darstellung auf den hinteren Bildschirmen. Der Projektor ist mit dem ersten Computer verbunden.

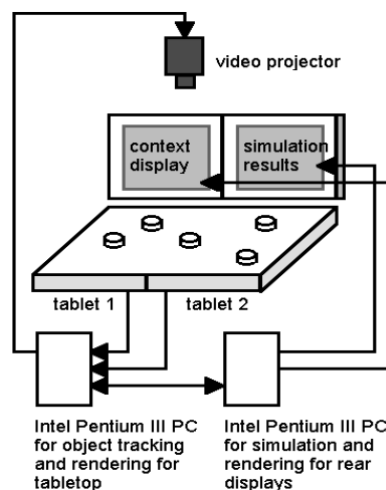


Abbildung 20 – Aufbau des SenseTable-Systems, bestehend aus zwei Tablet-Computern, zwei Desktop-Computern, zwei externen Bildschirmen sowie einem Projektor über dem Tisch. Quelle: [23]

Als Tangibles kommen *Pucks* zum Einsatz. Ein Puck ist ein zylinderartiges Objekt mit zwei seriellen Schnittstellen zum Anschließen von Koppellementen. Beispielsweise kann ein Drehregler an einen Puck gekoppelt werden, um einen Wert auslesen zu können. Pucks können mit grafischen Elementen verbunden werden und diese verschieben oder verändern. Die Bindung an ein Element erfolgt durch Annäherung. Aufgrund technischer Einschränkungen der Tablets können immer nur zwei Pucks gleichzeitig erkannt werden. Deshalb werden der Reihe nach Pucks für kurze Zeit abgeschaltet, um die Position der aktivierten Tangibles zu erkennen. Danach werden sie wieder aktiviert.



Abbildung 21 – Auf dem linken Bild ist ein Puck zu sehen, dessen obere zwei Anschlüsse mit einem Drehregler gekoppelt sind, während die unteren beiden noch frei sind. Im rechten Bild ist derselbe Puck zusehen mit US-Cent Münze zum Größenvergleich. Quelle: [23]

Als Demoanwendungen wurde unter anderem eine Lernsoftware zur Veranschaulichung chemischer Prozesse entwickelt. Der Nutzer kann mit den Pucks Atome und Moleküle erfassen und zusammenbringen, sodass die Kombinationen gegebenenfalls eine Reaktion auslösen und damit die Bildung neuer Moleküle visualisieren.

Powerwall Interactions

An der Universität Stuttgart wurde ein System entwickelt, welches das kollaborative Analysieren von Eye-Tracking-Daten auf mehreren Bildschirmen und mit einem Tangible User Interface ermöglicht. Über verbundene Geräte erfolgt die Interaktion und die Ausgabe der relevanten Daten kann auf verschiedene Displays (Tablets, Smartphones, Powerwalls, ...) verteilt werden, um größere Personengruppen an der Datenanalyse teilhaben zu lassen.

Jedes verbundene Gerät verfügt über einen sogenannten „privaten Raum“ und einen „öffentlichen Raum“. Der öffentliche Raum wird über alle angeschlossenen Geräte synchronisiert, während der private Raum nur für das jeweilige Gerät zur Verfügung steht. Visualisierungen zwischen diesen beiden Räumen können beliebig gewechselt werden. Der Moderator einer Analysesitzung kann in seinem privaten Raum Visualisierungen vorhalten, die zu passender Zeit den anderen Nutzern zur Verfügung gestellt werden. Um Gestenerkennung an der Powerwall zu ermöglichen, wurde ein Kinect-Sensor verwendet. Ein Microsoft PixelSense Tabletop-Computer kann mit Tangibles bedient werden, um geeignete Visualisierungen für die Daten auszuwählen. Die Analytiker können sich so in verschiedene Gruppen aufteilen, um verschiedene Aspekte gleichzeitig zu analysieren.

Das gesamte System kann über Gesten und Tangibles gesteuert werden. Das entwickelte Framework erlaubt plattformunabhängig das Hinzufügen von weiteren Geräten, die an der Sitzung teilnehmen. Ein zentraler Server speichert und synchronisiert Daten und regelt die Kommunikation zwischen den Geräten, um deren Rechen- und Speicherkapazität nicht zu belasten. Das Projekt wurde exemplarisch anhand eines Analyseszenarios mit Eye-Tracking-Daten und dem PixelSense-Tabletop auf einer über 12 Quadratmeter großen Powerwall erfolgreich ausgeführt. [24]

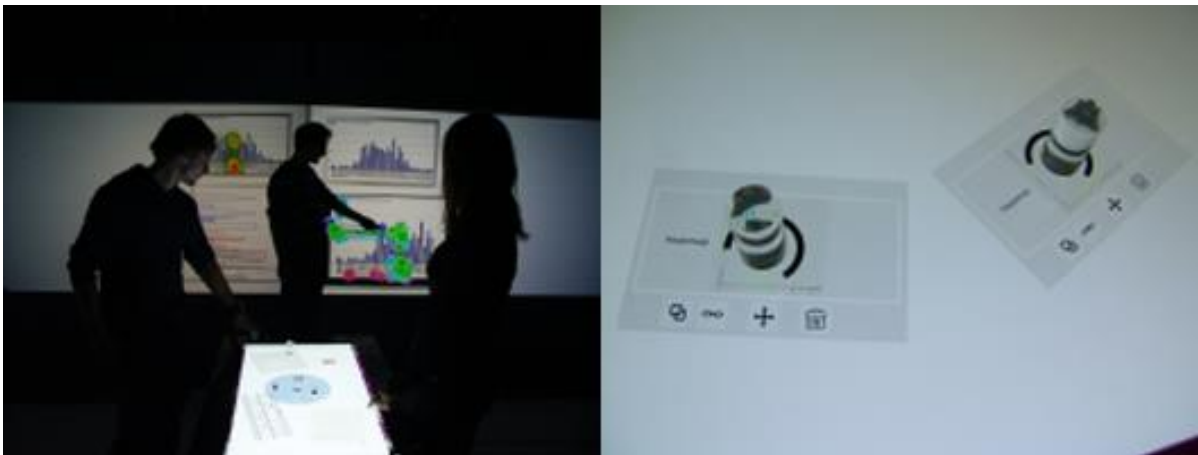


Abbildung 22 – Im linken Bild sind drei Nutzer zu sehen, die gleichzeitig Eye-Tracking-Daten mit Hilfe einer Powerwall und einem Tabletop-Computer analysieren. Im rechten Bild sind Tangibles mit virtuellen Objekten zu sehen, die auf dem Tabletop-Computer verwendet werden können, um Visualisierungen auszuwählen. Quelle: [24]

Für eine Studie wurden beispielhaft Daten von Urlaubshotels herangezogen, aus denen die Probandengruppen innerhalb einer vorgegebenen Zeit ein Hotel ihrer Wahl mit den Tangibles ermitteln mussten. Die Studienbeobachter stellten fest, dass den Nutzern das Interaktionskonzept Spaß machte, die Nutzer das Konzept schnell erlernen konnten und auch mit komplexeren logischen Ausdrücken zurechtkamen.

4 Aufgabenbeschreibung

In diesem Kapitel wird die Aufgabenbeschreibung genau erklärt, die dieser Arbeit zugrunde liegt. Kapitel 4.1 befasst sich mit dem Hintergrund der Aufgabe, Kapitel 4.2 beschreibt die Aufgabe an sich und Kapitel 4.3 legt dar, welche Schritte unternommen wurden, um zu einer Lösung zu gelangen.

4.1 Hintergrund

Eine typische Eye-Tracking-Datenanalyse wird aktuell so durchgeführt: Das Eye-Tracking-Datenanalyseteam versammelt sich um einen Computer, auf dem die Software zur Datenauswertung installiert ist. Ein Analyst übernimmt federführend die Kontrolle und bedient das System, während die anderen lediglich verbal eingreifen können. Die mangelnde Bildschirmgröße verhindert es dabei, mehrere Visualisierungen gleichzeitig anzuzeigen. Möchte man mehrere Darstellungen der Datensätze haben, sind weitere Bildschirme erforderlich. Eine Aufteilung der Analysten in mehrere Teams ist auch nur dann möglich, falls mehrere Computer zur Verfügung stehen. Die gemeinsame Datenanalyse im Team wird somit sehr umständlich und die Übersichtlichkeit leidet. Zur Analyse werden beispielsweise spezielle Software-Lösungen von Tobii [1] verwendet.

Zur Verbesserung der kollaborativen Datenanalyse wurden bereits zwei Diplomarbeiten durchgeführt, die Eye-Tracking-Datenanalysen mittels Tangible User Interfaces, Powerwalls und Gestensteuerung ermöglichen [26, 27]. Diese Arbeit soll nun das in den beiden vorausgegangenen Arbeiten in einer ersten Version erstellte Konzept erweitern, in dem ein Tangible User Interface zur Datenanalyse konzipiert wird, wobei der Fokus in Richtung der Tangibles verschoben ist. Im ersten Konzept der oben zitierten Arbeiten sind die Tangibles lediglich passiv. Sie besitzen keine Sensoren oder Kommunikationseinheiten und können auf dem Tabletop nur gesetzt, verschoben, gedreht oder entfernt werden. In dieser Arbeit liegt der Fokus auf *Active Tangibles*. Durch Sensoren, Displays, Schalter und Kommunikationsmodule werden die passiven Tangibles zu aktiven und können zusätzlich zu den passiven Eigenschaften (setzen, verschieben, drehen, entfernen) unter anderem Daten anzeigen, Daten ändern, Gesten interpretieren und mit anderen Geräten kommunizieren. Dies kann die Übersichtlichkeit der Darstellung für die Analysten erhöhen und bietet neue Interaktionsmodalitäten (etwa Drücken oder Schütteln), die mit dem ersten Konzept nicht realisierbar sind.

4.2 Aufgabenbeschreibung

Im Rahmen dieser Arbeit soll ein Interaktionskonzept entwickelt werden, mit dem unter Einsatz von Tabletop-Computern und Tangible User Interfaces beliebige Datensätze kollaborativ analysiert werden können. Zunächst soll dafür eine Recherche zu Grundlagenthemen und verwandten Arbeiten durchgeführt werden. Der nächste Schritt ist die Entwicklung eines Interaktionskonzepts, das die kollaborative Datenanalyse mit Tabletop-Computern und Tangible User Interfaces ermöglicht. Das Konzept soll in einen Prototyp implementiert werden, der am Beispiel einer realen Eye-Tracking-Datenanalyse getestet wird. Darüber hinaus soll eine Evaluation des Konzepts durch Studien stattfinden.

4.3 Vorgehensweise

Die Erstellung, Implementierung und Evaluation des Interaktionskonzepts orientiert sich am klassischen Wasserfallmodell in der Softwareentwicklung. Das Modell bietet gute Plan- und Kontrollmöglichkeiten und ist aufgrund der gegebenen Anforderungen sowie zeitlichen

Rahmenbedingungen gut einsetzbar [28]. Im Folgenden werden nun die einzelnen Phasen des Modells beschrieben.

Anforderungsanalyse

Zu Beginn wird eine Anforderungsanalyse durchgeführt, um das Konzept entwickeln zu können. Die Analyse umfasst die Beobachtung von Datenanalysen sowie Gespräche mit Analyseexperten, um deren Probleme oder Wünsche für künftige Systeme zu erfahren. Die Ergebnisse der Analyse sind in Kapitel 5 zu finden.

Entwicklung Interaktionskonzept

Basierend auf der Grundlagenrecherche und der Anforderungsanalyse wird ein Konzept entwickelt, um mit Tabletop-Computern und Tangibles Datenanalysen durchführen zu können. Die Beschreibung dieses ersten Konzepts ist in Kapitel 6.1 zu finden.

Szenario-Entwurf und Evaluation

Es wird in Zusammenarbeit mit Experten ein Szenario entworfen, das einer realen Eye-Tracking-Datenanalyse entspricht und die Grundlage für die Vorstudie bildet. Während einer Vorstudie mittels *Paper-Prototyping* [29] sollen Probanden anhand des Szenarios die erste Entwicklungsstufe des Interaktionskonzept testen. Das Ziel dieser Studie ist die Verbesserung des Interaktionskonzepts. Die Beschreibungen zur Vorstudie, deren Durchführung, deren Ergebnisse und entwickelten Szenarios sind in Kapitel 6.2 nachzulesen.

Erweiterung Interaktionskonzept

Basierend auf den Ergebnissen der Vorstudie wird das Interaktionskonzept überarbeitet und eine finale Version vorgestellt, die im Prototyp implementiert wird. Eine genaue Übersicht über die Änderungen sowie das finale Konzept sind in Kapitel 6.3 zu finden.

Implementierung des Prototyps

Auf Basis des entwickelten Interaktionskonzepts wird der Prototyp implementiert. Er gliedert sich in Hardware (selbstgebaute Tangibles) und Software (Tabletop und Tangibles). Zuerst werden die Tangibles entworfen und basierend darauf die Funktionen festgelegt, die in der Tabletop-Software enthalten sind. Die Entwicklung der Tangibles ist in Kapitel 7.1 und die Tabletop-Software in Kapitel 7.2 beschrieben.

Prototyp Evaluation

Der entwickelte Prototyp dient als Grundlage für die zweite Studie. In dieser Studie soll eine Gegenüberstellung des hier entwickelten Analysekonzepts mit dem typischen Analysekonzept einer Eye-Tracking-Datenanalyse eine Bewertung ermöglichen. Die Studienteilnehmer sind alle Experten auf dem Gebiet der Eye-Tracking-Datenanalyse. Die Ergebnisse der Prototyp Evaluation sind in Kapitel 8 zu finden.

5 Anforderungsanalyse

Zu Beginn ist eine Analyse der Anforderungen für das zu entwickelnde Konzept nötig. Die gesammelten Resultate werden in diesem Kapitel vorgestellt. Die Anforderungsanalyse dient dazu, das Konzept zu entwickeln und sie bildet eine Basis sowohl für den Test als auch für die Evaluation des Prototyps. Die Erfassung erfolgt durch Gespräche mit Experten und Beobachtungen bei aktuellen Eye-Tracking-Datenanalysevorgängen, um den Analyseprozess verbessern zu können. Eine Teilmenge der hier vorgestellten Anforderungen wird in das finale Konzept übernommen und im Prototyp umgesetzt. Kapitel 5.1 beschreibt die Anforderungen an das Interaktionskonzept, Kapitel 5.2 die Anforderungen an die Darstellung auf der Tabletop-Oberfläche, Kapitel 5.3 die Anforderungen an die Tangibles und Kapitel 5.4 die restlichen Anforderungen an das Interaktionskonzept, die nicht in eine Kategorie geordnet werden können.

5.1 Anforderungen an das Konzept

Im Folgenden werden die gesammelten Anforderungen an das System geschildert. Dabei handelt es sich hauptsächlich um *weiche Anforderungen* [28], die subjektiv ausgelegt werden und daher nicht jedem einzelnen Endbenutzer des Systems, welches das Interaktionskonzept implementiert, gerecht werden können.

Da Tangible User Interfaces nicht jedem bekannt sind, sollte das zu entwickelnde System einen einfachen Einstieg in die Bedienung bieten, möglichst intuitiv sein und ein logisches Bedienungskonzept verfolgen. Zusätzliche Hilfsfunktionen wie Informationstexte, Sprechblasen oder Signalfarben, die sich bei Bedarf abschalten lassen, sollen den Benutzer durch das System führen. Der Tabletop-Computer soll benutzt werden, um kollaboratives Arbeiten zu ermöglichen und um die Problematik der Einzelbedienung am Desktop-PC zu eliminieren. Jeder Analyst soll in den Analysevorgang eingreifen können, in dem die Tangibles verwendet werden. Das Arbeiten mit dem System soll Spaß machen.

Auch wenn der Fokus auf aktiven Tangibles liegt, soll die Touch-Interaktion des Tabletop-Computers nicht vernachlässigt werden. Das Konzept soll so ausgelegt sein, dass es zur Erreichung eines Zieles mehrere Wege gibt, die sowohl Touch- als auch Tangible-Interaktion beinhalten. So kann jeder Nutzer das von ihm vorgezogene Interaktionsschema bevorzugt einsetzen.

Das Konzept soll durch zwei Studien während und nach der Entwicklung überprüft werden. Eine erste Studie soll am Beispiel einer Eye-Tracking-Datenanalyse Schwachstellen des Interaktionskonzepts ermitteln, sodass diese bei der Entwicklung berücksichtigt werden können. Da das Konzept nicht für eine spezielle Zielgruppe entwickelt wird, sollen die Studienteilnehmer möglichst verschieden (Alter, Bildung, Vorkenntnisse, Geschlecht) sein, um bestmögliche Ergebnisse zu erhalten. Eine Möglichkeit zur Durchführung ist Paper-Prototyping [29]. Die zweite Studie soll einen Vergleich zwischen der Implementierung des Interaktionskonzepts im Prototyp und einem aktuell genutzten System zur Eye-Tracking-Datenanalyse ermöglichen.

5.2 Anforderungen an die Darstellung

Dieser Abschnitt beschreibt die Anforderungen an die Darstellung der Tabletop-Oberfläche.

Visualisierungsformen

Im Idealfall ersetzt das entwickelte System komplett die Analyse am Desktop-PC. Damit das gelingt, müssen natürlich auch die wichtigsten Visualisierungsformen für Daten umgesetzt werden, die das aktuelle Analyse-System unterstützt. Die Liste der wichtigsten Visualisierungsformen [30] umfasst folgende Elemente:

- *Heatmaps*
- *Scan-Path-Diagramme*
- *Gaze Duration Sequence-Diagramme*
- *Gaze Duration Distribution-Diagramme*
- *Fixation Point-Diagramme*

Im entwickelten Interaktionskonzept, welches im Prototyp implementiert werden Heatmaps und Scan-Path-Diagramme integriert. Jedoch können mit wenig Aufwand auch die anderen Visualisierungsformen eingebunden werden. Eine Visualisierung soll eine beliebige Anzahl von Probandendatensätzen anzeigen können und soll sich automatisch aktualisieren, wenn die Probandendatensätze geändert werden. Zur Hervorhebung von Datensätzen kann eine Highlight-Funktion benutzt werden, die beispielsweise in Scan-Path-Diagrammen die entsprechenden Datensätze mit Signalfarben versieht und die restlichen etwas transparenter darstellt. Dies wird der Prototyp alles ermöglichen.

Parametrisierung von Visualisierungen

Darüber hinaus ist eine Parametrisierung der Anzeige geplant. Für jede Visualisierung sollen bestimmte Eigenschaften einstellbar sein, um die Analyse zu unterstützen. So würde etwa die farbliche Hervorhebung eines Probanden innerhalb einer Gruppe in einem Scan-Path-Diagramm die Lesbarkeit erhöhen. Ein anderer Ansatz wäre, die Achsenskalen auf einen Bereich einzuschränken, sodass Daten ausgeblendet werden, die für den aktuellen Analyseschritt nicht relevant sind. Die Gaze Duration Sequence- und Gaze Duration Distribution-Diagramme können hinsichtlich der Zahl der dargestellten *Areas of Interest* (AOIs) [30] oder auf der vertikalen Zeitachse eingeschränkt werden. Idealerweise lässt sich eine Parametrisierung an mehrere Visualisierungen koppeln, sodass in jeder dieselben Datenbereiche angezeigt werden. Weitere gewünschte Funktionen sollen das Zoomen auf einen bestimmten Bereich innerhalb einer Visualisierung sowie Filter- und Sortierfunktionen für Gruppen von Datensätzen sein. Ein Beispiel hierfür wäre die Sortierung aller zugewiesenen Datensätze eines Diagramms aufsteigend nach deren benötigter Zeit für eine bestimmte Szenarioaufgabe, wobei Probanden herausgefiltert werden sollen, die länger als 30 Sekunden gebraucht haben.

Das finale Interaktionskonzept wird über je eine Funktion zum Filtern und Hervorheben verfügen, wobei als Parameter Probandendatensätze angegeben werden. So ist etwa das Filtern von drei Datensätzen aus sieben möglich, aber nicht das Filtern von Probanden, die älter als 30 Jahre sind.

Externe Bildschirme

Der Anzeigebereich soll nicht ausschließlich auf den Tabletop-Computer beschränkt sein, sondern sich erweitern lassen. Der verwendete Computer PixelSense von Microsoft verfügt über eine Full HD Auflösung mit 1920 x 1080 Pixeln bei einer Bildschirmdiagonale von knapp 100 cm. Ab einer bestimmten Anzahl an dargestellten Elementen kann es nötig sein, Elemente wieder zu entfernen oder aber die Oberfläche zu erweitern. Eine mögliche Umsetzung hierfür soll das Einbinden von externen Monitoren, Powerwalls, Tablet PCs und Smartphones sein. Ein Analyst kann auf diese Weise mit seinem Tablet die Oberfläche erweitern, in dem er sich damit in die Nähe des Tabletops bewegt. Nun kann er etwa per Wischgeste oder über ein Tangible eine Visualisierung in seinen Bereich verschieben und sich wieder vom Tabletop entfernen und an anderer Stelle mit der Visualisierung arbeiten. Es soll auch möglich sein, Elemente von Tablets oder Smartphones auf den Tabletop-Computer zu schieben. Metainformationen zu Probandendaten eignen sich ebenfalls dafür, auf externen Geräten angezeigt zu werden.

Der Prototyp wird über eine Funktion verfügen, um externe Bildschirme, die an den Tabletop-Computer angeschlossen sind, in die Darstellung von Visualisierungen oder Metadaten einzubeziehen.

Nutzererkennung

Der Tabletop-Computer sollte die Ausrichtung seiner Menüelemente dem Benutzer anpassen. Am Rand des Tisches können Annäherungssensoren platziert werden, die reagieren, wenn sich ein Benutzer dem Tisch nähert und alle virtuellen graphischen Elemente auf der Tabletop-Oberfläche in seine Richtung ausrichten. Erkennt das System mehr als einen Benutzer, sollte es die Möglichkeit geben, die Elemente manuell entsprechend seinen Bedürfnissen auszurichten (rotieren, verschieben, skalieren). Mittels *Radio Frequency Identification*-Sendern (RFID), die etwa in Zugangsberechtigungskarten integriert sind, kann die Benutzererkennung anstelle der Annäherungssensoren durchgeführt werden.

Wird das Tabletop-System von einer Aufgabe stark beansprucht, sollte ein Indikator den Benutzern dies verdeutlichen. Ein schmaler roter Rand am Bildschirm des Tabletops könnte rot aufleuchten, wenn das System ausgelastet ist oder die Oberfläche aktuell nicht reagiert. Alternativ ist auch eine virtuelle Sanduhr denkbar, die in diesem Fall eingeblendet wird und aus früheren Windows Betriebssystemen bekannt ist. So hätten die Benutzer immer eine Rückmeldung über den aktuellen Systemzustand.

Der Prototyp wird den Nutzer bei rechenintensiven Aufgaben darüber informieren. Dies tritt gewöhnlich auf, wenn Visualisierungen gerendert werden, da dies mit steigender Anzahl der Probandendatensätze länger dauert. Eine Ausrichtung der Menüelemente oder Nutzeridentifizierung ist aus Zeitgründen nicht vorgesehen.

Sitzungsspeicherung

Im Zusammenhang mit der Nutzererkennung kann auch eine Speicherung der Analysesitzung ermöglicht werden. Hierzu merkt sich das Tabletop-System den Benutzer anhand seines RFID-Tags sowie die Anzahl und Einstellungen der Tangibles, die in die Richtung des Benutzers ausgerichtet sind, sobald er den Tisch verlässt. Andere Benutzer des Systems können nun zusätzlich zu ihren eigenen Tangibles auch die desjenigen benutzen, der sich gerade vom Tisch entfernt hat. Kommt der betroffene Nutzer später wieder zurück, begrüßt ihn ein virtueller Assistent mit einer Meldung auf der Tabletop-Oberfläche oder alternativ per Audio-Nachricht. Der Assistent gibt Anweisungen, welche Tangibles der Nutzer wo auf

der Oberfläche platzieren muss, damit der letzte Stand der Sitzung wiederhergestellt werden kann. Die Einstellungen der Tangibles, die Verbindungen zwischen ihnen und die zuvor vorhandenen Visualisierungen werden vom System wiederhergestellt und der Nutzer kann mit der Analyse fortfahren. Eine nutzerunabhängige Sitzungsspeicherung würde keine Nutzeridentifizierung benötigen. Beim Start des Systems könnte der Assistent fragen, ob der letzte Stand der vorherigen Sitzung wiederhergestellt oder eine neue gestartet werden soll. Der Prototyp wird über keine Möglichkeit zur Speicherung und Wiederherstellung einer Analysesitzung verfügen.

5.3 Anforderungen an die Tangibles

Die Tangibles sind das zentrale Steuerelement eines Tangible User Interfaces. Im Zusammenspiel mit der Multi-Touch-Oberfläche des Tabletop-Computers können sie die mangelhafte Kollaborativität bei Analysen am Desktop-PC deutlich verbessern. Im Fokus dieser Arbeit stehen intelligente Tangibles (active Tangibles), die dem Nutzer einen Mehrwert bieten. Deswegen soll ein Informationsaustausch zwischen Tabletop und Tangible möglich sein. Die Tangibles sollen darüber hinaus mit Sensoren ausgestattet werden und sich auf ihre Umgebung einstellen können.

Es gibt mehrere Ansätze, um mit Tangibles zu arbeiten. Ein Ansatz wäre, dass jedes Tangible nur eine Funktion (*kollaborativer Ansatz*) realisiert (etwa Datenspeicher oder Ausgabe einer Visualisierung). Ein alternativer Ansatz wäre, dass jedes Tangible individuell (*individueller Ansatz*) arbeitet und dass es schrittweise mit Informationen und Funktionen bestückt wird. Im Folgenden werden nun beide Ansätze näher beschrieben.

Kollaborativer Ansatz

Werden die Tangibles nun miteinander verbunden, können Daten ausgetauscht und gegebenenfalls angezeigt werden. Der Vorteil hierbei ist die strikte Aufgabentrennung der Tangibles, was zu einer erhöhten Übersichtlichkeit führen kann. Allerdings stehen bei diesem Ansatz mehr Objekte auf dem Tabletop-Computer, was den Platzverbrauch erhöht. Das Zuweisen und Löschen von Daten sowie die Funktionsauswahl eines Tangibles kann über ein Auswahlmenü auf der Tabletop-Oberfläche erfolgen. Das Koppeln zweier Tangibles kann durch ein nahes Zusammenstellen der Tangibles oder durch das Platzieren des einen Tangibles auf einem speziell markierten Punkt (Andockpunkt) auf der Tabletop-Oberfläche des anderen Tangibles erreicht werden. Es ist auch denkbar, die Umriss eines Tangibles auf der Oberfläche darzustellen und eine Verbindung zu initiieren, wenn sich ein weiteres Tangible in diesen Umrissen befindet. Zum Entkoppeln von Tangibles kann eines der Tangibles kurz von der Oberfläche angehoben werden oder der Nutzer wählt durch Berührung einen entsprechenden Menüpunkt in einem virtuellen Menü des betreffenden Tangibles auf der Tabletop-Oberfläche.

Individueller Ansatz

Ein anderer Ansatz wäre, dass jedes Tangible individuell arbeitet und dass es schrittweise mit Informationen und Funktionen bestückt wird. Beispielsweise könnten dem Tangible zuerst Daten und danach eine Visualisierung zugeordnet werden, wodurch diese mit den zugeordneten Daten generiert und angezeigt wird. Somit wird weniger Platz verbraucht, da weniger Tangibles wie beim anderen Ansatz für dieselbe Aufgabe benötigt werden. Jedoch leidet die Übersichtlichkeit, da es nun nicht mehr so offensichtlich ist, welche Daten wie im Tangible gespeichert sind. Ein Beispiel hierzu:

Ein einzelnes Tangible ist auf dem Tabletop platziert und in einem Scan-Path-Diagramm werden fünf Probandendatensätze angezeigt. Da es nur ein Tangible gibt, muss das Scan-Path-Diagramm offensichtlich von diesem Tangible generiert worden sein. Allerdings kann die Darstellung von genau fünf Probandendatensätzen mehrere Ursachen haben. Es können beispielsweise dem Tangible zuerst über den Menüpunkt Data Source 15 Probandendatensätze zugewiesen worden sein, wovon anschließend über den Menüpunkt Filter zehn Probandendatensätze gefiltert wurden. Das resultierende Scan-Path-Diagramm würde fünf Scan-Paths anzeigen. Es kann aber auch sein, dass dem Tangible einfach nur fünf Probandendatensätze über den Menüpunkt zugewiesen wurden. Die Ausgabe ist in beiden Fällen identisch, aber es ist nicht eindeutig, wie sie zustande kommt.

Die Daten- und Funktionszuweisung kann analog wie beim ersten Ansatz erfolgen, das Koppeln und Entkoppeln von Tangibles entfällt bei diesem Ansatz. Im Prototyp wird der kollaborative Ansatz verwendet, da bei einer gemeinschaftlichen Analyse auch Tangibles unter den Analysten ausgetauscht werden können und es für jeden Analyst immer ersichtlich ist, welche Funktion das Tangible momentan realisiert und welche Daten ihm zugewiesen sind.

Generische Tangibles

Um den größtmöglichen Nutzen von den Tangibles zu haben, sollen sie generisch einsetzbar und nicht zweckgebunden sein. Deswegen sollen die Tangibles auf kleinem Raum viele Komponenten (Sensoren, Schalter, Knöpfe, Kommunikationsmodule,...) integrieren, die bei der Datenanalyse hilfreich sein können. Im Folgenden werden einige Komponenten sowie deren Nutzen tabellarisch dargestellt, die für die Konstruktion von Tangibles in Frage kommen können (siehe Tabelle 1).

Komponente	Eigenschaften
Bildschirm	Text- und Grafikausgabe
Lautsprecher	Audioausgabe
Mikrofon	Audioeingabe
Gyroskop, Beschleunigungssensor	Gestenerkennung
RFID Sender	Benutzererkennung
Bluetooth, WLAN	Kommunikation
Druckknopf, Schalter	Zustandsänderung von Eigenschaften
Schiebe-, Drehregler	Wertänderung von Eigenschaften
Leuchtdiode	Statusanzeige
Batterie, Stromanschluss	Stromversorgung

Tabelle 1 – Auflistung möglicher Komponenten und deren Eigenschaften, die in ein Tangible eingebaut werden können.

Zweckgebundene Tangibles

Zusätzlich zu generischen Tangibles können weitere entwickelt werden, die zweckgebunden und somit nicht für alle Datenanalyseszenarien passend sind. Ein Beispiel hierfür ist ein Slider-Tangible, welches beispielsweise zur Parametrisierung von Visualisierungen eingesetzt werden kann. Wird es mit einer Visualisierung verknüpft, so kann es die Achsenskalierungen modifizieren. Mit zwei Slidern ist auch das Eingrenzen der Skalierung auf einen bestimmten Bereich denkbar. So könnten bei Scan-Path-Diagrammen die Anzahl der Fixationen pro Proband etwa auf einen Maximalwert festgelegt oder bei Gaze Duration Sequenzdiagrammen die Anzahl der Area of Interests (AOI) limitiert werden.

Ein Sortier-Tangible kann die Probandendaten nach bestimmten Kategorien, welche aus dem Datensatz der Probanden abgeleitet werden sortieren. Mittels einer Schüttelgeste kann der Analyst durch die Kategorien wechseln, wobei die aktuelle textuell auf einem Bildschirm

dargestellt wird. Durch eine schnelle Drehung oder durch Schütteln des Tangibles kann die Sortierreihenfolge von auf- in absteigend oder umgekehrt geändert werden. Es lassen sich somit die Probanden etwa nach ihrem Alter oder der benötigten Zeit für eine Aufgabe sortieren.

Auch eine Art digitale Füllstandsanzeige ist angedacht: Sie wird auf einem Bereich der Visualisierung platziert und stellt einen relativen Wert grafisch dar. So ist es denkbar, dass auf einer Heatmap die Menge der Fixationen in einem kleinen Bereich angezeigt oder die Menge der Sakkaden in einem Scan-Path-Diagramm dargestellt wird.

Ein Stempel-Tangible kann verwendet werden, um von einer Visualisierung oder der gesamten Tabletop-Oberfläche einen Screenshot zu erzeugen. Die Screenshots landen in einem Verzeichnis auf dem Tabletop-Computer und über einen externen Monitor können bis zu vier Bilder gleichzeitig dargestellt werden. Auch ein fortlaufender Wechsel der Bilder ist denkbar. Mit diesem Tangible lässt sich die Dokumentation eines Analysevorgangs erreichen.

Einbindung von Smartphones und Tablets

Eine weitere Idee ist die Verwendung von Smartphones und Tablets zum Fotografieren von Visualisierungen, die dann auf dem hochauflösenden Display der Geräte genauer betrachtet werden können. Der Benutzer fotografiert mit dem externen Gerät eine Visualisierung und kann sie dann auf dem Gerät betrachten, sodass mehr Raum auf der Tabletop-Oberfläche für andere Dinge zur Verfügung steht. Elemente lassen sich auch auf externe Geräte verschieben, in dem eine Wischgeste in Richtung des Geräts gemacht wird, wenn es sich in der Nähe des Tabletop-Computers befindet. Dabei wird eine Kopie der Daten (Metadaten von analysierten Studienteilnehmern, Visualisierungen) an das Gerät gesendet, welches sie darstellen kann.

Außer der Anbindung eines externen Monitors, der an den Tabletop-Computer angeschlossen ist, unterstützt der Prototyp keine weiteren externen Geräte wie Smartphones oder Tablets. Natürlich ist es möglich, die Tabletop-Oberfläche abzufotografieren, um etwa die Zoom-Funktion eines Smartphones zur Vergrößerung einer dargestellten Visualisierung zu benutzen.

Energiesparmaßnahmen

Da die Tangibles eine eigene Stromversorgung besitzen und möglichst lange einsatzbereit sein sollen, können Energiesparmaßnahmen umgesetzt werden. Tangibles sollen automatisch erkennen, wenn sie nicht gebraucht werden und Komponenten abschalten, um Strom zu sparen. Außerdem soll es möglich sein, die Batterie von Tangibles über einen Stromversorgungsanschluss aufzuladen oder sie möglichst einfach zu ersetzen. Die Tangibles besitzen keine Stromsparmechanismen, sondern verfügen lediglich über einen Ein- und Ausschalter.

5.4 Sonstige Anforderungen

Weitere Anforderungen sind unter anderem das Speichern eines Screenshots einer Visualisierung, um die Analyse in Bildern dokumentieren zu können. Dies soll explizit durch den Nutzer möglich sein oder periodisch geschehen (zeitgesteuert und bei der Änderung eines Datensatzes). Unterstützend dazu ist eine Notizfunktion angedacht, um für Datensätze eine Kurzbeschreibung speichern zu können. Eine Eingabe könnte mittels virtueller Tastatur auf der Tabletop-Oberfläche oder per Audio-Aufnahme erfolgen. Ein Tangible könnte mit Mikrofonen ausgestattet und als Diktiergerät benutzt werden. Mittels der integrierten

Spracherkennung von Windows 7 wäre sogar eine Umwandlung in eine Textnotiz denkbar. Der Prototyp wird keine der in diesem Abschnitt genannten Vorschläge enthalten, da aus Zeitgründen nicht alle Ideen umgesetzt werden können.

6 Lösungskonzept

In diesem Kapitel wird das Interaktionskonzept vorgestellt, welches im Rahmen dieser Arbeit entwickelt wird. Basierend auf den ermittelten Anforderungen (siehe Kapitel 5) werden in Abschnitt 6.1 Vorüberlegungen als Basis des Interaktionskonzepts dargestellt. Es ist keine komplette Neuentwicklung, sondern es kombiniert verschiedene in verwandten Arbeiten (siehe Kapitel 3) benutzte Ansätze, um ein generisches Interaktionskonzept zur Datenanalyse zu erstellen. Eine anschließende Benutzerstudie (Abschnitt 6.2) soll das Interaktionskonzept diskutieren und verbessern. Die Studienergebnisse fließen direkt in die Gestaltung des finalen Konzepts ein, welches in einem Prototyp implementiert wird. Abschnitt 6.3 beschreibt das finale Konzept inklusive der Änderungen durch die Benutzerstudie.

6.1 Vorüberlegungen

Die erste Version des Interaktionskonzepts orientiert sich an vergleichbaren Arbeiten (siehe Kapitel 3), an der Aufgabenstellung (siehe Kapitel 4.2) und an den Anforderungen (siehe Kapitel 5), die gesammelt wurden. Folgende Fragen werden in den folgenden Abschnitten detailliert erläutert:

- Welche Tangible-Typen gibt es?
- Wie funktionieren die Tangibles?
- Wie können Tangibles kombiniert werden?
- Wie sieht die Tabletop-Oberfläche aus?
- Wie arbeiten Tangibles und Tabletop-Computer zusammen?
- Wie werden externe Monitore eingebunden?

Tangible-Typen

Es wird drei verschiedene Tangible-Typen geben (generischer Typ, Notiz-Tangible-Typ und Monitor-Tangible-Typ). Skizzen der Typen sind in Abbildung 25 zu sehen. Der erste Typ ist generisch einsetzbar, besitzt einen Bildschirm an der Oberseite, einen seitlichen Druckknopf, einen Beschleunigungssensor, eine Leuchtdiode und kommuniziert per Bluetooth. Ihm wird zur Laufzeit seine eigentliche Funktion zugewiesen. Dieser Typ ist der Haupttyp und wird bei Analysen vorrangig verwendet werden, da er flexibler und leistungsfähiger als die anderen beiden Typen ist. Der zweite Typ wird als Notiz-Tangible bezeichnet und dient dazu, eine Kurznotiz in Textform zu verfassen, die einem Probanden oder einer Gruppe von Probanden zugeordnet wird. Damit können zur Laufzeit Informationen zu bestimmten Aspekten der Analyse gespeichert werden, auf die im weiteren Verlauf oder bei folgenden Analysevorgängen zurückgegriffen werden kann. Der letzte Typ wird als Monitor-Tangible bezeichnet. Er dient dazu, Visualisierungen oder Metainformation von Probanden auf einem externen Gerät (Smartphone, Tablet, Powerwall, externer Monitor) anzuzeigen. Die Funktionen der beiden letzteren Typen sind von Beginn an festgelegt.

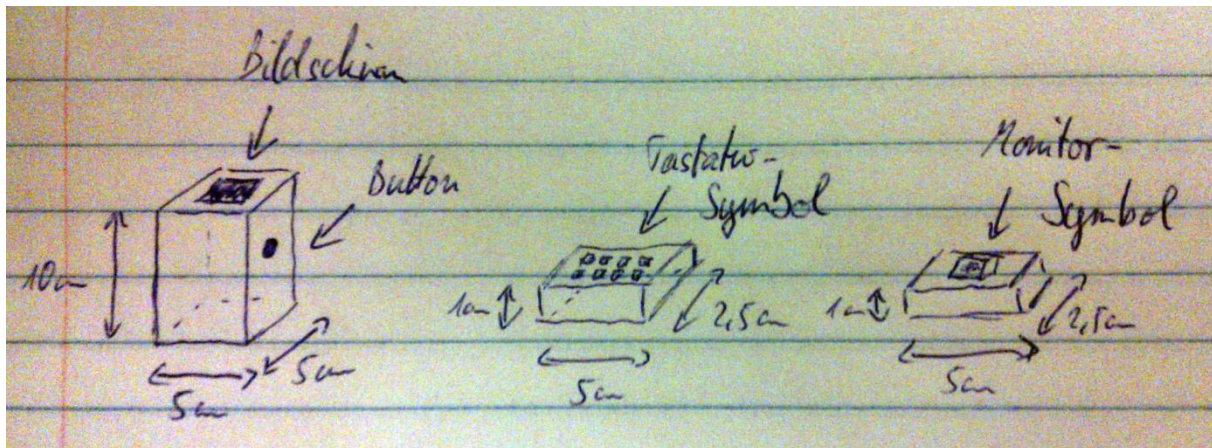


Abbildung 25 – Skizzen der drei geplanten Tangible-Typen in der ersten Version des Interaktionskonzepts. Generisches Tangible mit Druckknopf und Bildschirm auf der Oberseite (links), Notiz-Tangible zum Erfassen einer Notiz für einen Proband oder eine Gruppe von Probanden (mittig) und Monitor-Tangible zum Darstellen von Visualisierungen oder Metadaten von Probandendatensätzen auf einem externen Monitor.

Funktionszuweisung

Um ein generisches Tangible mit einer Funktion belegen zu können, erscheint ein Auswahlmenü auf der Tabletop-Oberfläche, wenn das Tangible zum ersten Mal auf den Tisch gestellt wird (siehe Abbildung 26). Hier bieten sich nun vier Unterpunkte zur Auswahl an:

- *Probandendaten*
- *Visualisierung*
- *Filter*
- *Hervorhebung*

Tippt der Nutzer ein Element an, erscheint ein weiteres Menü, in welchem konkrete Einstellungen gesetzt werden können. Basierend auf der Auswahl wird dem Tangible eine Kategorie zugewiesen. Wählt der Nutzer *Probandendaten* aus, ist das Tangible ein Input-Element. Wählt er hingegen *Visualisierung* aus, wird es zum Output-Element. Die anderen beiden Menüpunkte führen zu einem Modifier-Element.

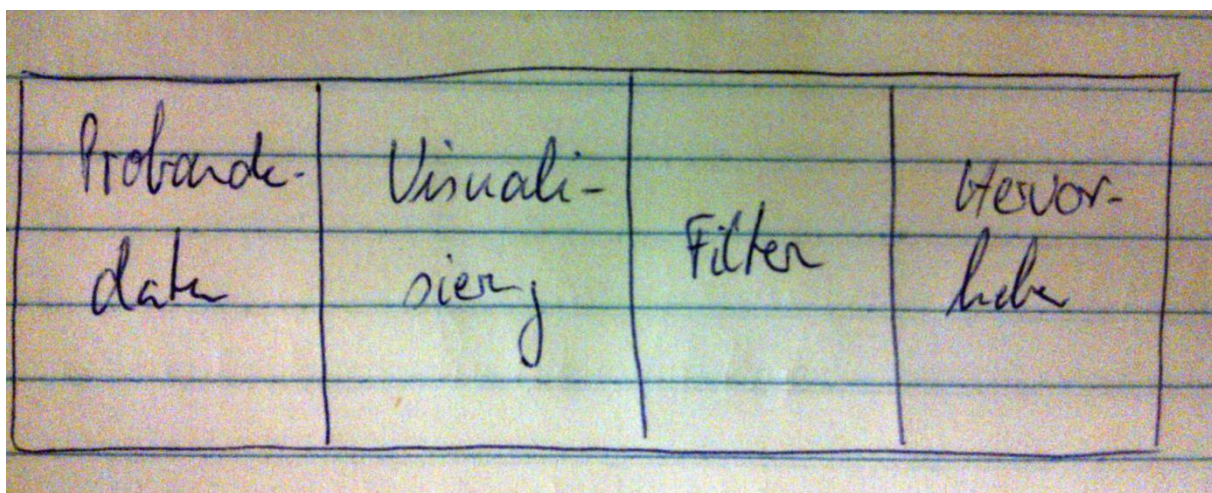


Abbildung 26 – Skizze des Auswahlmenüs der ersten Version des Interaktionskonzepts. Die Auswahl eines Menüpunktes öffnet ein Untermenü zum Setzen von konkreten Einstellungen bezüglich der gewählten Kategorie. Selektierte Menüpunkte sind farblich hervorgehoben.

Tangible-Verkettung

In anderen Arbeiten ([22], [25]) zu diesem Thema wurden Tangibles virtuell miteinander verbunden, um das Ergebnis (eine Visualisierung oder eine Berechnung) zu beeinflussen. Dieses Prinzip erweist sich als interessanter Ansatz für das zu entwickelnde Interaktionskonzept, sodass er auch im Rahmen dieser Arbeit verwendet wird. Da ein Ziel des Interaktionskonzepts die Möglichkeit der gemeinschaftlichen Datenanalyse ist, ist es sinnvoll, die zu analysierenden Daten aufteilen und kombinieren zu können. Dies kann über das Zuweisen von Daten an Tangibles und Verbinden von Tangibles erreicht werden. Deshalb werden sie wie bereits im Abschnitt *Tangible-Typen* erwähnt in drei Kategorien (Input, Modifier, Output) eingeteilt, die maßgebend für die Kopplung sind. Eine Kette von Tangibles besteht aus einem Input-Element, einer beliebigen Anzahl von Modifier-Elementen und einem Output-Element. Die Daten fließen immer vom Input zum Output (siehe Abbildung 27).

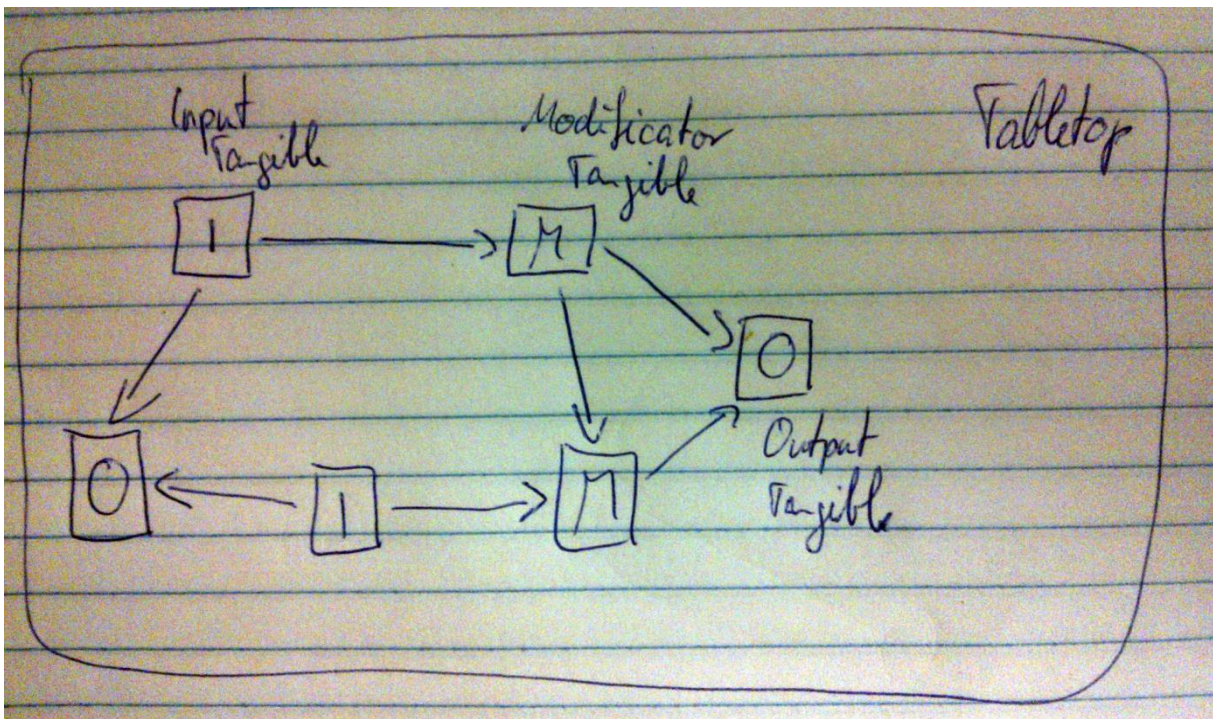


Abbildung 27 – Skizze einer möglichen Verkettung von Tangibles in der ersten Version des Interaktionskonzepts. Auf der Tabletop-Oberfläche sind sechs Tangibles platziert, je zwei Input-, Modifier- und Output-Tangibles. Die Verbindungspfeile deuten die Datenflussrichtung an. Die Daten fließen immer vom Input zum Output. Input-Tangibles speisen Daten in das System ein. Modifier-Tangibles empfangen Daten, verändern diese gegebenenfalls und leiten sie weiter. Output-Tangibles geben eingehende Daten in einer Visualisierung aus.

Sind zwei Tangibles miteinander kompatibel (zum Beispiel ein Input-Element und ein Output-Element), können sie verbunden werden. Tabelle 2 zeigt eine vollständige Kompatibilitätstabelle. Eine Verbindung zwischen zwei Tangibles wird hergestellt, wenn sie sich in genügend kleinem Abstand zueinander befinden. Sie wird auf dem Tabletop-Computer dargestellt. Um eine Verbindung wieder zu lösen, muss der Abstand vergrößert werden. Das Konzept ähnelt den Filter Flow Graphen [31].

Kompatibel?	Input	Modifier	Output
Input	Nein	Ja	Ja
Modifier	Ja	Ja	Ja
Output	Ja	Ja	Nein

Tabelle 2 – Die Tabelle zeigt eine Übersicht darüber, welche Tangible-Kategorien mit welchen kompatibel sind. Kompatibel bedeutet, dass Tangibles dieser Kategorien verbunden werden können. Die Datenflussrichtung wird automatisch anhand der Regel (Input -> Modifier -> Output) bestimmt.

Tangible-Eigenschaften

Ein Tangible kann immer nur eine Funktionalität realisieren und soll im Zusammenspiel mit anderen Tangibles verwendet werden. Die Funktionalität kann jederzeit geändert werden, in dem durch das Drücken des Druckknopfs das Menü aufgerufen wird, um die Einstellungen des Tangibles zu ändern. Um die Einstellungen komplett zu löschen, ist neben dem Weg über das Menü auch eine Schüttelgeste vorgesehen, die das Tangible schnell wieder in den Ausgangszustand zurückversetzt.

Der Bildschirm des generischen Tangibles zeigt Informationen darüber an, welche Funktion es aktuell besitzt. Sind dem Tangible beispielsweise Probandendaten zugewiesen, erscheint deren Anzahl im Display. Auf dem Bildschirm eines Output-Tangibles wird der Name der Visualisierung, bei einer Hervorhebung oder Filterung die Anzahl der hervorgehobenen oder gefilterten Elemente angezeigt. Die Zuweisung bleibt bei allen Tangibles erhalten, wenn das Tangible vom Tisch genommen wird.

Tabletop-Oberfläche

Die Tabletop-Oberfläche soll unterstützend agieren und dem Nutzer Interaktionsmöglichkeiten anbieten. Sie zeigt etwa das Auswahlmenü eines Tangibles oder die Datenvisualisierung in Form einer Heatmap an. Daneben zeigt sie die Verbindungen zwischen einzelnen Tangibles sowie Informationen an, die zu groß für den Bildschirm des Tangibles sind. Zwischen dem Tabletop-Computer und den Tangibles besteht eine Abhängigkeit; der eine kann ohne den anderen nicht sinnvoll arbeiten.

Externer Monitor

Ein externer Monitor kann zur Erweiterung der Benutzeroberfläche an den Tabletop-Computer angeschlossen werden. Mit dem Monitor-Tangible können entweder Visualisierungen in vergrößerter Form (wenn es mit einem Output-Tangible verbunden wird) oder Metadaten von Probanden (wenn es mit einem Input- oder Modifier-Tangible verbunden wird) angezeigt werden. Sind mehrere Monitor-Tangibles aktiv, wird die Oberfläche des externen Monitors gleichmäßig aufgeteilt. Die Anzahl der einzelnen Teilbereiche des Bildschirms entspricht der Anzahl der verbundenen Monitor-Tangibles.

6.2 Vorstudie

In einer Vorstudie soll untersucht werden, wie die Lösungsansätze der Aufgabenstellung von Probanden unterschiedlicher Qualifikationen sowie Vorkenntnissen und verschiedenen Alters nachvollzogen werden können. Zur Verifikation wird eine Studie anhand des Paper-Prototypings [29] ausgearbeitet und mit Probanden durchgeführt. Diese Art der Studie eignet sich auch bereits, wenn noch kein fertiges System zur Verfügung steht, sondern nur ein Konzept vorhanden ist.

Das Ziel dieser Vorstudie ist die Verifikation und Verbesserung der ersten Version des Interaktionskonzepts, welches in Abschnitt 6.1 beschrieben ist. Die Ergebnisse der Studie fließen in die Entwicklung des finalen Interaktionskonzepts ein, welches im Prototyp implementiert wird.

Die Testpersonen erhalten exemplarische Aufgaben, wie sie auch im finalen System zu absolvieren sein werden und versuchen diese mit den Papierskizzen der Benutzeroberfläche durchzuführen. Idealerweise wird für jede Aktion ein entsprechendes Element der Oberfläche als Papierskizze hergestellt und bei Bedarf dem Nutzer vorgelegt. So kann dieser mit dem System interagieren und dieses bewerten. Alle Testpersonen müssen zudem zu Beginn und am Ende der Studie einen Fragebogen ausfüllen, um die Bewertung der Studie zu verbessern. Der erste Fragebogen ermittelt die Vorkenntnisse der Teilnehmer in den Bereichen Smartphones, Tablets, Tabletop-Computern und Tangible User Interfaces. Der zweite Fragebogen fragt nach generellen Eindrücken der Studie. Beispielsweise:

- Sind die Hilfestellungen ausreichend?
- Ist der Einstieg in das System schwierig?
- Sind die Tangibles selbsterklärend?

Die Fragebögen sind im Anhang A und B zu finden. Die Probanden sind zudem angehalten, sämtliche Gedanken laut auszusprechen sowie eigene Ideen, Fehler und Anmerkungen vorzutragen.

Zur Durchführung stehen Tangibles zur Verfügung, die aus Pappe hergestellt wurden. Zusätzlich wird ein normaler Tisch benötigt, der in etwa die Größe eines Tabletop-Computers hat. Ein externer Monitor kann durch eine Wand oder ein Whiteboard simuliert werden.

Im Folgenden wird nun die Aufgabenstellung für die Studie vorgestellt. Dabei durchläuft jeder Proband ein Szenario aus acht Aufgaben, die aufeinander aufbauen und typischerweise während der Analyse von Eye-Tracking-Daten ausgeführt werden müssen (siehe Abschnitt 4.1). Sollte ein Proband eine Aufgabe alleine nicht lösen können, wird Hilfestellung gewährt, um die weiteren Aufgaben bearbeiten zu können. Die Aufgaben sind so gestellt, dass im Allgemeinen nur eine Lösung vorhanden ist. Ob diese Lösung die sinnvollste ist, soll sich durch die Studie zeigen. Der Aufgabenablauf ist so konzipiert, dass die Probanden zuerst einfache Dinge wie das Erzeugen einer Visualisierung erreichen und im weiteren Verlauf fortgeschrittene Konzepte wie Filtern oder Speichern einer Notiz einsetzen. Im Folgenden werden die einzelnen Aufgaben kurz vorgestellt.

1) Ein Tangible mit drei Probandendatensätzen verbinden

Hierbei soll der Nutzer ein Tangible auf dem Tisch platzieren und im sich öffnenden Auswahlmenü drei beliebige Probandendatensätze auswählen sowie das Menü durch einen Knopfdruck am Tangible schließen. Die Daten sind nun im Tangible (*Probandentangible*) gespeichert und die Beschriftung des Displays deutet dies an (siehe Abbildung 28).

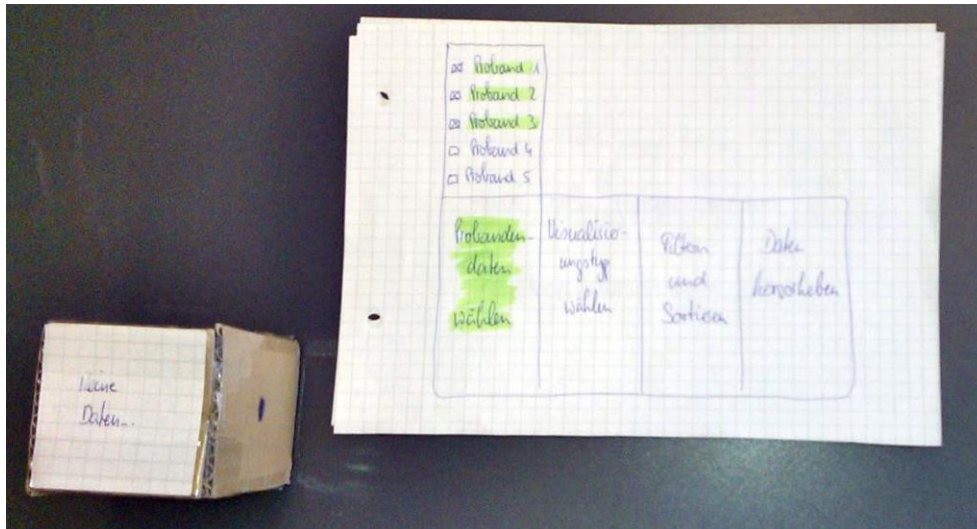


Abbildung 28 – Aufgabe 1 der Vorstudie: Zu sehen ist eine Papierskizze eines Tangibles mit Bildschirm auf der Oberseite und Druckknopf an der rechten Seite (links) sowie eines Auswahlmenüs (rechts). Grün hinterlegte Menüpunkte sind selektiert

2) Selektierte Probandendaten in einer Heatmap-Visualisierung ausgeben

Bei der zweiten Aufgabe soll der Nutzer ein weiteres Tangible auf dem Tisch platzieren und ihm eine Heatmap-Visualisierung zuweisen. Anschließend müssen beide Tangibles nahe zueinander platziert werden, sodass ein Verbindungspfeil und eine Heatmap erscheinen (siehe Abbildung 29).

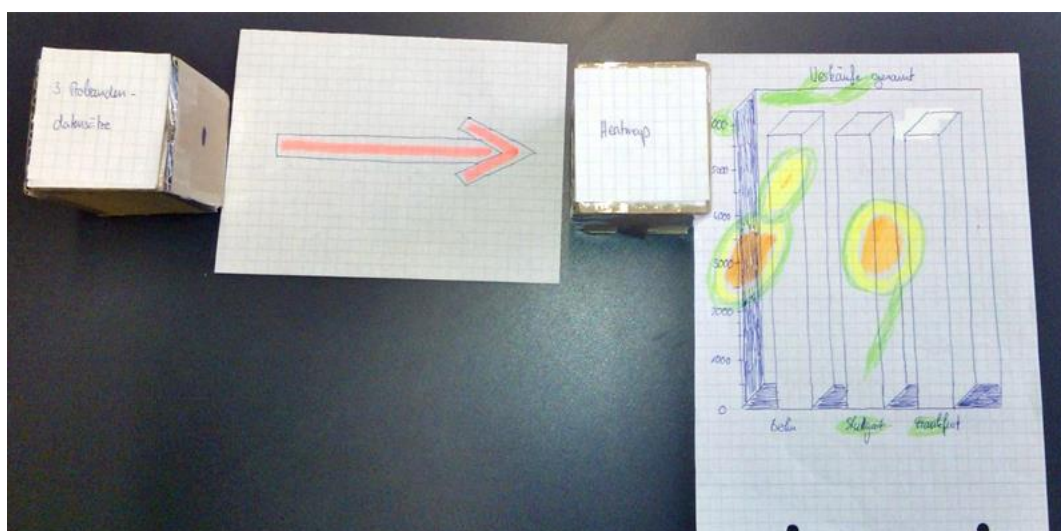


Abbildung 29 – Aufgabe 2 der Vorstudie: Zu sehen sind Papierskizzen von zwei Tangibles, die durch einen Pfeil verbunden sind. Die Daten werden in einer Heatmap ausgegeben (rechts). Der Pfeil symbolisiert die Datenflussrichtung.

3) *Selektierte Probandendaten zusätzlich in einem Scan-Path-Diagramm ausgeben*

Zusätzlich zur Heatmap-Visualisierung soll nun ein Scan-Path-Diagramm angezeigt werden, welches dieselben Daten wie die Heatmap darstellt. Dazu muss der Proband ein weiteres Tangible auf den Tisch stellen, es mit einer Scan-Path-Visualisierung wie in der Aufgabe zuvor verknüpfen und dieses Tangible in die Nähe des ersten Tangibles verschieben. Es werden wieder ein Verbindungspfeil sowie ein Scan-Path-Diagramm angezeigt (siehe Abbildung 30).

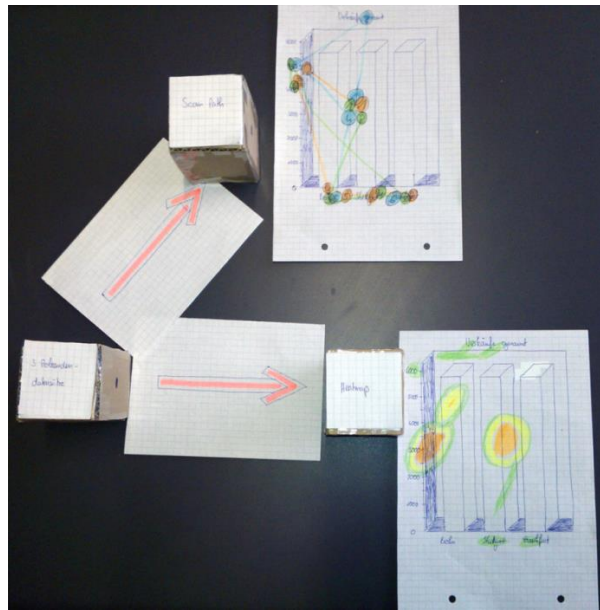


Abbildung 30 – Aufgabe 3 der Vorstudie: Zu sehen sind drei Tangibles, zwei Verbindungspfeile und zwei Visualisierungen. Die Daten des Tangibles auf der linken Seite werden sowohl in einer Heatmap als auch in einem Scan-Path-Diagramm ausgegeben.

4) *Zwei Datensätze filtern, sodass jeweils nur noch ein Datensatz angezeigt wird*

Ein viertes Tangible muss in der Mitte der drei Tangibles platziert werden, sodass ein Eingangspfeil vom Probandentangible erscheint und zwei Ausgangspfeile in Richtung der Visualisierungstangibles. Im Auswahlmenü müssen unter dem Punkt „Filtern und Sortieren“ zwei der drei Datensätze ausgewählt werden, sodass sie in den Visualisierungen nicht angezeigt werden (siehe Abbildung 31).

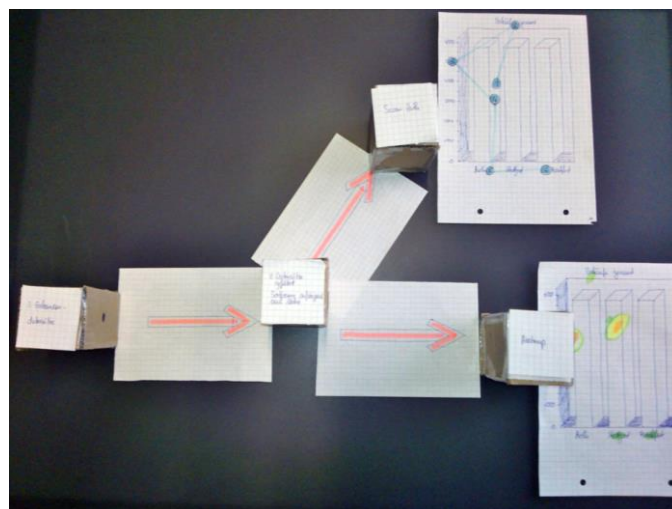


Abbildung 31 – Aufgabe 4 der Vorstudie: Zu sehen sind Papierskizzen des Filterprinzips. Das zweite Tangible (Filter) von links filtert zwei von drei Datensätzen, die vom Probandentangible ganz links zur Verfügung gestellt werden. In den Visualisierungen, die mit dem Filter verbunden sind, wird jeweils nur ein Datensatz angezeigt.

5) Filter-Tangible-Einstellungen löschen und Tangible entfernen

Das Löschen von Verknüpfungen mit einem Tangible kann entweder über eine Schüttelgeste oder durch Entfernen der Selektion(-en) im Auswahlmenü erfolgen. Nach dem Entfernen des Tangibles verschwinden alle Verbindungspeile sowie die Visualisierungen und es sind noch drei voneinander entkoppelte Tangibles auf dem Tisch vorhanden.

6) Mit dem Heatmap-Tangible einen Datensatz hervorheben

Der Proband soll das Tangible benutzen, das die Heatmap (*Heatmap-Tangible*) darstellt, und dessen Funktionalität ändern, sodass ein beliebiger Datensatz im Scan-Path-Diagramm hervorgehoben wird (*Highlight-Tangible*). Um im Auswahlmenü unter dem Punkt „Highlight“ Probandendaten zum Hervorheben zur Auswahl zu haben, muss das Tangible allerdings zuerst in der Nähe des Probandentangibles platziert werden. Am Ende dieser Aufgabe steht eine Kette aus Probanden-, Highlight- und Visualisierungstangibles auf dem Tisch (siehe Abbildung 32).

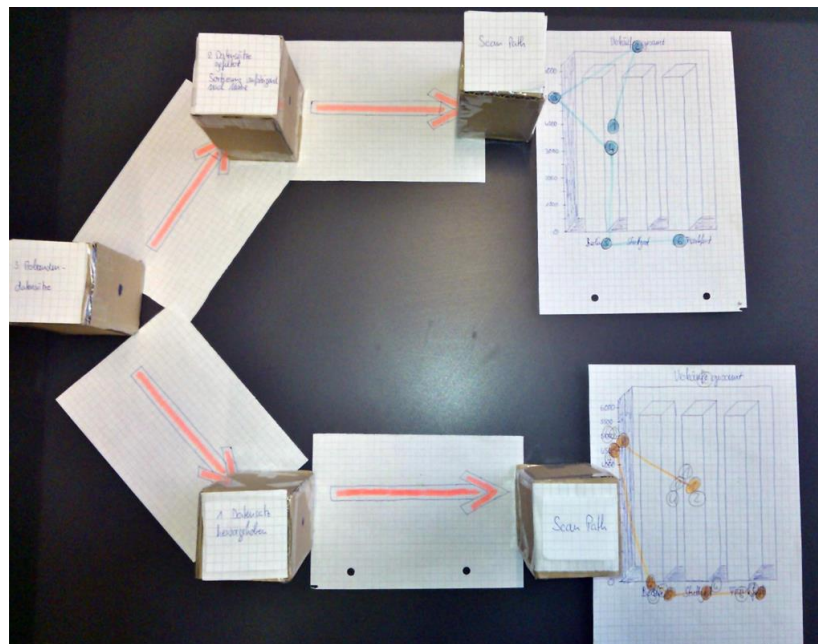


Abbildung 32 – Aufgabe 6 der Vorstudie: Zu sehen sind Papierskizzen zum Filtern und Hervorheben von Probandendatensätzen. Der obere Pfad filtert zwei von drei Datensätzen und zeigt die entsprechende Scan-Path-Visualisierung an, der untere Pfad hebt einen von drei Datensätzen hervor und zeigt ebenfalls die entsprechende Scan-Path-Visualisierung an. Beim Hervorheben werden nicht selektierte Datensätze mit mehr Transparenz, beim Filtern überhaupt nicht dargestellt.

7) *Metadaten der Probanden auf einem externen Bildschirm anzeigen*

Hierfür muss der Proband nur das entsprechende Tangible (Monitor-Tangible) in der Nähe des Probandentangibles platzieren, damit die Metadaten auf dem externen Monitor dargestellt werden (siehe Abbildung 33).

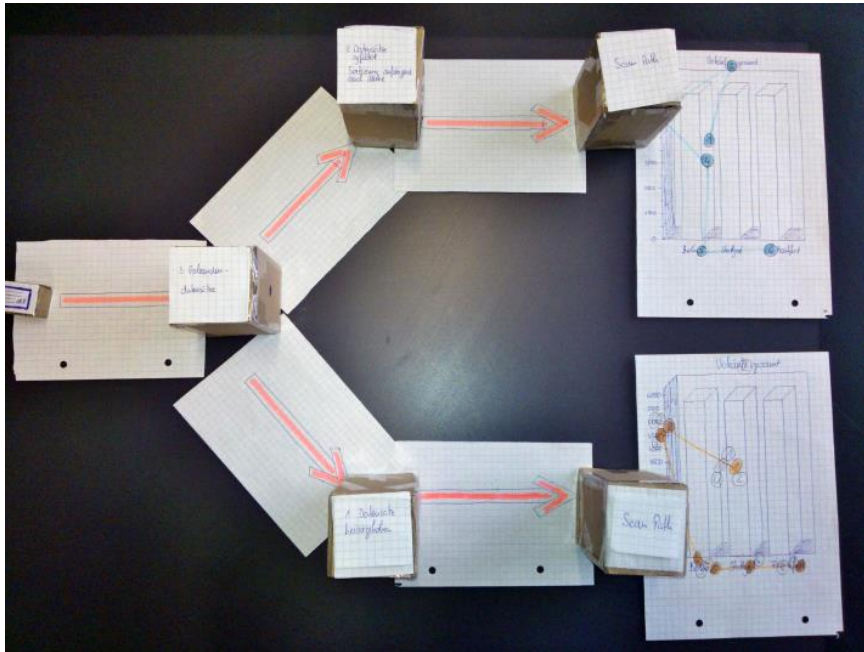


Abbildung 33 – Aufgabe 7 der Vorstudie: Zu sehen ist eine Papierskizze des Monitor-Tangibles (links). Das Monitor-Tangible zeigt die Metadaten der Probandendatensätze auf einem externen Bildschirm an (nicht im Bild). Statt eines Verbindungspfeils ist hier nur eine Verbindungslinie dargestellt.

8) *Notiz zu den aktuellen Probandendatensätzen speichern*

Das Notiz-Tangible muss neben einem Probandentangible platziert werden, sodass mit Hilfe der erscheinenden virtuellen Tastatur kann eine Kurznotiz verfasst oder, falls bereits vorhanden, bearbeitet werden (siehe Abbildung 34).

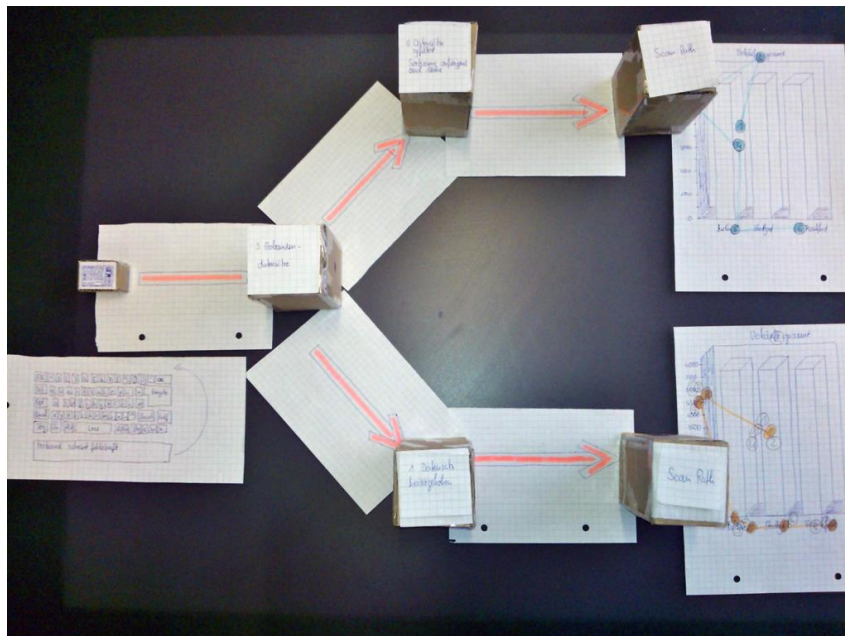


Abbildung 34 – Aufgabe 8 der Vorstudie: Zu sehen ist eine Papierskizze des Notiz-Tangibles (links). Das Notiz-Tangible öffnet eine virtuelle Tastatur zum Verfassen einer Notiz, welche für alle verbundenen Probandendatensätze gespeichert wird. Statt eines Verbindungspfeils ist hier nur eine Verbindungslinie dargestellt.

6.2.1 Durchführung

Die Studie wird mit insgesamt acht Teilnehmern durchgeführt. 75% davon sind Mitarbeiter des Instituts Visualisierung und Interaktive Systeme (VIS) oder Studenten, die an diesem Institut arbeiten oder dort eine Studien- oder Abschlussarbeit schreiben. Eine Auflistung aller Teilnehmer sowie deren Vorkenntnisse sind in Tabelle 3 zu finden. Die folgenden Abschnitte beschreiben kurz den Studienablauf der einzelnen Probanden.

	Geschlecht	Alter	Beruf	Eye-Tracking-Datenanalyse?	Erfahrung mit Touch-Geräten?	Erfahrung mit Tabletop-Computern?	Erfahrung mit Tangibles?
1	m	27	Stud. Mit.	Fortgeschritten	Ja	Ja	Ja
2	w	24	Student	Anfänger	Ja	Nein	Nein
3	m	25	Student	Experte	Ja	Ja	Ja
4	m	25	Student	Fortgeschritten	Ja	Ja	Ja
5	m	30	Wiss. Mit.	Anfänger	Ja	Ja	Ja
6	m	30	Student	Anfänger	Ja	Ja	Nein
7	m	27	Polizist	Anfänger	Ja	Nein	Nein
8	m	29	Wiss. Mit.	Experte	Ja	Ja	Ja

Tabelle 3 – Tabellarische Darstellung der Vorkenntnisse der Studienteilnehmer basierend auf deren Angaben.

Proband 1

Proband 1 ist männlich, 27 Jahre alt und Mitarbeiter des VIS. Er hat bereits mit Tangibles und Tabletop-Computern gearbeitet und besitzt ebenfalls Kenntnisse in der Analyse von Eye-Tracking-Daten. Zusätzlich verfügt er über umfassende Kenntnisse in der Bedienung von Systemen mit berührungsempfindlichen Bildschirmen. Bei der Bearbeitung der Aufgaben ist er sehr kreativ und findet teilweise auch eigene Lösungswege für Aufgaben. Insgesamt beurteilt er das System als einfach zu bedienen, bemängelt aber hauptsächlich fehlende Hilfsfunktionen für den Nutzer. Die Bedienung mit Tangibles macht ihm Spaß, allerdings findet er deren Nutzung nicht sehr intuitiv. Er würde sich weitere Systeme wünschen, die über ein Tangible User Interface verfügen, sofern das Anwendungsszenario es zulässt.

Probandin 2

Probandin 2 ist weiblich, 24 Jahre alt und studiert Chemie. Sie verfügt über keinerlei Kenntnisse zu Tabletops, Tangibles und Eye-Tracking-Datenanalysen, ist jedoch sehr vertraut mit der Bedienung von Smartphones. Die ersten Aufgaben bereiten der Probandin einige Verständnisprobleme, die sich im Verlauf der Studie deutlich verringern. Sie findet den Einstieg in das System schwierig, freundet sich aber mit dem Bedienkonzept an. Auch sie bemängelt die unzureichenden Hilfsfunktionen. Das Arbeiten macht ihr allerdings großen Spaß.

Proband 3

Proband 3 fällt dadurch auf, dass er abgesehen von der ersten Aufgabe keinerlei Probleme mit dem System hat und das Szenario ohne Hilfe komplett durchspielen kann. Er ist 25, männlich und Diplomand beim VIS. Dank seiner umfangreichen Vorkenntnisse in den Bereichen Smartphones, Tablets, Tabletop-Computern und Tangibles liefert er wertvolle Hilfen und alternative Lösungen zur Erreichung des Aufgabenziels. Das System gefällt ihm, er arbeitet gerne damit. Gleichzeitig fehlen ihm aber auch einige Hilfsfunktionen für Nutzer mit weniger Vorwissen.

Proband 4

Proband 4 besitzt wie Proband 3 umfangreiche Vorkenntnisse in den Bereichen Smartphones, Tablets, Tabletop-Computern und Tangible User Interfaces. Jedoch fällt ihm die Bedienung des Systems wesentlich schwerer. Auch er ist ein männlicher 25-jähriger Student. Der Einstieg gelingt ihm nicht so einfach, die Tangibles bereiten ihm etwas Probleme und die Hilfen des Systems sind ihm zu unklar. Er arbeitet gerne mit dem System, ist aber nicht davon überzeugt, dass ähnliche Bedienkonzepte in anderen Anwendungsgebieten sinnvoll wären.

Proband 5

Proband 5 kennt sich nicht mit Eye-Tracking-Daten aus, jedoch mit Tablets, Tabletops, Tangible und Natural User Interfaces. Er ist 30 Jahre, arbeitet beim VIS und findet äußerst viele alternative Lösungswege, um die Aufgaben zu erledigen. Er arbeitet zügig, hat keine Schwierigkeiten, das Interaktionskonzept zu erlernen und ist insgesamt ein Fan der Tangible User Interfaces. Trotzdem merkt auch er die mangelhaften Hilfsfunktionen an.

Proband 6

Proband 6 liefert keine neuen Ideen während der Studie, bestätigt aber einige bereits aufgetretene. Auch er ist Student mit umfangreichen Kenntnissen in den Bereichen NUI und TUI. Auch mit Tabletop-Computern hat er bereits gearbeitet. Ihm missfallen die unzureichenden Hilfen, die große Anzahl an zu benutzenden Tangibles sowie der schwierige Einstieg in die Bedienung des Systems. Wie Proband 4 ist auch er nicht davon überzeugt, dass Tangible User Interfaces in anderen Anwendungsfällen von Vorteil sind.

Proband 7

Proband 7 ist der einzige Teilnehmer der Studie, der nicht studiert. Er ist 27 Jahre alt und Polizist. Seine Vorkenntnisse umfassen die Bedienung von Smartphones, wohingegen er noch nie mit Tablets, Tabletops oder Tangibles gearbeitet hat. Die anfängliche Skepsis wandelt sich mit zunehmender Studiendauer in Freude, als er das Bedienkonzept durchschaut hat. Wie alle Teilnehmer würde auch er sich mehr Hilfestellungen wünschen, die vor allem die anfänglichen Hürden verringern sollten. Auch er findet Gefallen an der Bedienung mit Tangibles und würde sich weitere Systeme dieser Art im Alltag wünschen.

Proband 8

Proband 8 ist Experte in Sachen Eye-Tracking-Datenanalysen, hat an mehreren Projekten mit Tangible User Interfaces teilgenommen und zeigt während der Studie viele kreative Vorschläge, die das Konzept benutzerfreundlicher gestalten können. Nach kurzer Eingewöhnungsphase hat er keine Probleme, das System zu durchschauen und es durch eigene Vorschläge zu erweitern. Er wünscht sich, dass das System jederzeit dem Nutzer die aktuellen Möglichkeiten präsentiert, sodass dieser besser geleitet wird. Abschließend empfiehlt er noch die Lektüre anderer TUI-Systeme, um weitere Ideen zu bekommen.

6.2.2 Zusammenfassung der Studienergebnisse

Die Ziele der Vorstudie bestanden hauptsächlich darin, zu evaluieren, ob das entwickelte Interaktionskonzept für andere Benutzer mit unterschiedlichen Vorkenntnissen nachvollziehbar ist. Durch Benutzen von für den Teilnehmer intuitiven Aktionen sollte gezeigt werden, dass das Konzept leicht erlernbar und generisch für verschiedene Aufgabentypen einsetzbar ist. Dafür wurde eine Auswertung anhand der Daten aus den Fragebögen der Benutzer vorgenommen. Die Zusammenfassung ist in der nachfolgenden Tabelle zu sehen.

Aufgabenbeschreibung	\bar{x}	σ
War das Lösen der Aufgaben einfach (1) oder kompliziert (10)?	2,875	1,25
War es einfach (1) oder schwer (10), einen Einstieg in die Bedienung des Systems zu finden?	4,125	1,25
War es einfach (1) oder schwer (10), das Bedienkonzept des Systems zu verstehen?	3	2,07
Waren die Hilfestellungen des Systems ausreichend (1) oder nicht ausreichend (10)?	3,75	2,38
War die Funktionsweise der Tangibles leicht (1) oder schwer (10) ersichtlich?	2,75	2,05
War die Anzahl der zu benutzenden Tangibles angemessen (5)?	5,25	0,71
Hat das Arbeiten mit dem System Spaß (10) gemacht oder nicht (1)?	9,125	0,83
Würden Sie gerne auch andere Systeme mit Tangibles bedienen (10) oder nicht (1)?	7,375	3,50

Tabelle 4 – Auswertung der Fragebögen der Benutzerstudie, die die Probanden nach Beendigung der Aufgaben ausgefüllt haben. Der erste Wert gibt den Durchschnittswert aller Angaben an, der zweite Wert ist die Standardabweichung. Eine niedrige Standardabweichung bedeutet, dass die Antworten ähnlich sind.

Aus den erfassten Daten können wertvolle Schlussfolgerungen für das zu entwickelnde System abgeleitet werden. Die Probanden kamen generell gut mit den gestellten Aufgaben zurecht und konnten diese recht einfach lösen. Die Einstiegsbarriere erschien aber vielen zu hoch, was sich auch in der Bewertung niederschlägt. Hier sind weitere Hilfsfunktionen wie zum Beispiel aussagekräftige Meldungen während der Interaktion eine mögliche Lösung. Das Bedienkonzept im Gesamten schien leicht erlernbar, wobei die ersten Aufgaben meistens größere Probleme als die letzteren bereiteten. Auch dieser Punkt deutet darauf hin, dass mit mehr und besseren Hilfestellungen die Benutzer von Anfang an weniger Probleme haben könnten. Insgesamt sind die Probanden mit dem Konzept zufrieden, weil es einfach zu erlernen ist, es viele Facetten der Eye-Tracking-Datenanalyse abdeckt und weil es Spaß macht. Der letzte Punkt deckt sich mit [32], die ebenfalls zu dem Ergebnis kommen, dass Probanden lieber und effizienter mit Tangible User Interfaces als mit Touch Interfaces arbeiten. Auf die Frage, ob die Probanden gerne mehr Systeme mit Tangibles bedienen würden, kamen jedoch sehr geteilte Antworten, da sich nicht jedes System für Tangible-Interaktion eignen würde.

Daneben war ein weiteres Studienziel die Sammlung von Ideen, um das Bedienkonzept zu erweitern, zu verbessern und verständlicher zu gestalten. Hierfür sind die Anmerkungen und Gedanken der Teilnehmer während der Studie maßgeblich. Die folgende Tabelle 5 zeigt eine nach Nennungen sortierte Liste der am häufigsten angemerkteten Verbesserungsvorschläge. Die Liste beinhaltet nur Vorschläge, die von mindestens zwei Probanden unabhängig voneinander angegeben wurden. Im Anschluss werden die Vorschläge detailliert erläutert.

Nr.	Vorschlag	#
1	Buttonklick zur Bestätigung und zum Schließen des Menüs unnötig	8
2	Sinnvollere Hilfemeldungen („Stell mich auf den Tisch“ oder „Benutz mich“)	8
3	Meldung, wenn bestehende Daten in einem Tangible verändert werden	6
4	Tangible-Display sollte sich automatisch aktualisieren	3
5	Einstellungen eines Tangibles sollten kopiert werden können	3
6	Radius für Verbindung zwischen Tangibles anzeigen	3
7	Löschfeld im Menü zum Löschen der Zuweisungen eines Tangibles	3
8	Löschbutton zum Löschen der Zuweisungen eines Tangibles	3
9	Funktion zum Erfassen einer Notiz ins Menü aufnehmen	3
10	Möglichkeit anbieten, Screenshot einer Visualisierung speichern	2
11	Verbinden von Tangibles durch kurzes Berühren, danach größerer Abstand möglich	2
12	Audio-Aufnahme der Notiz anbieten	2
13	Tangible Umriss auf Tabletop darstellen, wenn weiteres Tangible benötigt wird	2

Tabelle 5 – Liste gewünschter Änderungen durch die Studienteilnehmer. In der Liste sind 13 Vorschläge enthalten, die von mindestens zwei Teilnehmern der Benutzerstudie unabhängig voneinander genannt wurden. Sie ist absteigend nach der Anzahl der Nennungen (letzte Spalte) eines Vorschlags sortiert.

1) Buttonklick zur Bestätigung und zum Schließen des Menüs unnötig

Der ursprüngliche Gedanke bestand darin, dass ein Nutzer das Tangible-Menü schließt, indem er den seitlichen Druckknopf am Tangible betätigt. Alle Probanden wiesen jedoch darauf hin, dass ihnen dieser Schritt zu umständlich sei und sie gerne eine alternative Lösung hätten. Ein möglicher Ansatz wäre, dass das Menü nach einer gewissen Zeit automatisch ausgeblendet wird.

2) Sinnvollere Hilfemeldungen („Stell mich auf den Tisch“ oder „Benutz mich“)

Alle Probanden bemängelten die Hilfemeldungen auf dem Tabletop-Computer und der Anzeige des Tangibles, da sie zu wenig aussagekräftig seien und vor allem Erstnutzern mit wenigen Vorkenntnissen keine Hilfe seien. Darum sollten bessere und häufigere Hilfemeldungen im finalen Prototyp enthalten sein. Als Starthilfe könnte etwa eine Sprechblase auf der Tabletop-Oberfläche den Nutzer auffordern, ein Tangible auf den Tisch zu stellen.

3) Meldung, wenn bestehende Daten in einem Tangible verändert werden

Sind einem Tangible bereits Daten zugewiesen (etwa eine Visualisierung) und der Nutzer ändert diese (er weist dem Tangible etwa vier Probandendatensätze zu), erscheint keine Bestätigungsmeldung, das hier Daten überschrieben werden. Viele Probanden störten sich daran, weil sie den Vorgang unabsichtlich ausführten und wünschten sich daher eine Änderung. Ein Bestätigungsdialo g könnte den Nutzer auf den Vorgang aufmerksam machen und unabsichtliche Änderungen verhindern.

4) Tangible-Display sollte sich automatisch aktualisieren

Die Anzeige des Tangibles sollte sich ursprünglich erst nach dem Schließen des Menüs aktualisieren. Jedoch findet fast die Hälfte der Probanden es informativer, wenn die Anzeige sofort bei einer Änderung reagieren würde. Somit wäre die Informationsdarstellung auf der Tabletop-Oberfläche und dem Bildschirm des Tangibles stets synchron.

5) Einstellungen eines Tangibles sollten kopiert werden können

Einige Nutzer wünschten sich eine Kopierfunktion für Tangibles. Dies könnte realisiert werden, in dem ein Tangible ohne Kategorie direkt neben einem anderen Tangible platziert wird, welches bereits eine Kategorie (und gegebenenfalls weitere Einstellungen) zugewiesen bekommen hat. Das „leere“ Tangible kopiert die Einstellungen des anderen Tangibles.

6) Radius für Verbindung zwischen Tangibles anzeigen

Drei von acht Probanden fänden es hilfreich, wenn Tangibles einen virtuellen Kreis als Orientierung um sich herum hätten, in welchem sie sich mit anderen Tangibles verbinden können. Wird ein weiteres Tangible innerhalb dieses Kreises platziert, können sich die beiden verbinden.

7) Löschfeld im Menü zum Löschen der Zuweisungen eines Tangibles

Zusätzlich zur Schüttelgeste wünschten sich einige Teilnehmer ein Löschfeld im Tangible-Menü, um die Einstellungen und Zuweisungen zu entfernen. Dies würde zum Paradigma passen, dass für alle Aktionen mehrere Lösungsmöglichkeiten vorgesehen sind.

8) Löschbutton zum Löschen der Zuweisungen eines Tangibles

Eine weitere alternative Löschmethode wäre ein separater Löschbutton am Tangible oder die Benutzung des bereits vorhandenen Buttons, indem dieser einfach lange gedrückt wird, um die Löschaktion auszuführen.

9) Funktion zum Erfassen einer Notiz ins Menü aufnehmen

Statt einem separaten Tangible kam die Idee auf, die Notizfunktion direkt ins Tangible-Menü zu integrieren und das extra Tangible einzusparen.

10) Möglichkeit anbieten, Screenshot einer Visualisierung speichern

Zusätzlich zur Anzeige auf externen Monitoren sollte es die Möglichkeit geben, von Visualisierungen Screenshots anzufertigen. Zur Umsetzung dieser Maßnahme sind mehrere Vorschläge angegeben worden. Ein Stempel-Tangible könnte beim Stempeln einer Visualisierung einen Screenshot speichern oder der Tabletop speichert regelmäßig Bildschirmfotos, um den Analysefortschritt zu dokumentieren.

11) Verbinden von Tangibles durch kurzes Berühren, danach größerer Abstand möglich

Das Verbinden von Tangibles soll nicht in einem bestimmten Radius erfolgen, sondern explizit, wenn der Nutzer beide Tangible direkt nebeneinander platziert. Anschließend können beide frei auf dem Tisch verschoben werden, die Verbindung bleibt erhalten. Diese Interaktionsform kollidiert mit dem Kopieren von Tangible-Einstellungen (Punkt 5). Es können nicht beide in dieser Form realisiert werden.

12) Audio-Aufnahme für Notiz anbieten

Da einige Probanden nicht gerne auf virtuellen Tastaturen tippen, kam der Vorschlag auf, eine Audio-Notiz speichern zu können. Zusätzlich dazu könnte die Notiz anschließend in digitalen Text umgewandelt und gegebenenfalls per (virtueller) Tastatur korrigiert werden. Das Notiz-Tangible kann mit einem Mikrofon ausgestattet und als Diktiergerät benutzt werden. Der Analyst nimmt das Tangible in die Hand und die Aufnahme beginnt. Stellt er das Tangible wieder auf den Tisch, stoppt die Aufnahme.

13) Tangible Umriss auf Tabletop darstellen, wenn weiteres Tangible benötigt wird

Um schneller Einblick in das Bedienkonzept erhalten zu können, kam der Vorschlag auf, dass die Umriss eines Tangibles auf dem Tabletop angezeigt werden sollen, wenn ein bereits mit Daten assoziiertes Tangible ein weiteres für eine Verbindung benötigt.

Mithilfe dieser Vorschläge wird das Interaktionskonzept aus Abschnitt 6.1 erweitert. Die konkreten Einflüsse der Studienergebnisse sind im Abschnitt 6.2.3 nachzulesen und das finale Konzept wird in Abschnitt 6.3 beschrieben.

6.2.3 Einfluss der Ergebnisse auf das Konzept

In diesem Abschnitt werden die Einflüsse der Studienergebnisse auf das Interaktionskonzept, das in Abschnitt 6.1 beschrieben ist, vorgestellt. Die hier vorgestellten Änderungen beeinflussen das finale Interaktionskonzept, welches im Prototyp umgesetzt wird. Der Prototyp wird in Kapitel 7 beschrieben.

Die Vorstudie zeigte, dass das bisherige Konzept vor allem bei Nutzern mit geringen Vorkenntnissen in den Bereichen Tabletop-Computern und Tangible User Interfaces über Schwächen verfügt. Deshalb sollen gerade für diese Nutzergruppe mehr Hilfen angeboten werden. Die Tabletop-Oberfläche wird eine Hinweismeldung anzeigen, die den Nutzer auffordert, ein Tangible auf den Tisch zu stellen, wenn der Tisch leer ist. Somit wird der Einstieg erleichtert, falls der Nutzer mit dem Einstieg in das Bedienkonzept überhaupt nicht zurechtkommt.

Sobald ein Tangible auf dem Tisch steht, erscheinen die Umriss eines zweitens Tangibles (*Verbindungsbereich*) neben dem gerade platzierten Tangible mit der textuellen Aufforderung, für eine Verbindung ein weiteres Tangible in den Umrissen zu platzieren. Das Verbinden zweier Tangibles erfolgt nur noch über das Platzieren innerhalb des angezeigten Bereichs. Wenn mehrere Tangibles auf der Tischoberfläche platziert sind und ein Benutzer eines dieser Tangibles verschiebt, so werden die Verbindungsbereiche kompatibler Tangibles grün markiert, um anzudeuten, dass eine Verbindung möglich wäre. Inkompatible Tangibles werden dagegen rot dargestellt.

Das virtuelle Auswahlmü des generischen Tangibles schließt sich einige Sekunden nach der letzten Interaktion mit dem Benutzer selbstständig. Da allen Probanden der explizite Druck auf den seitlichen Druckknopf am Tangible zu umständlich war, geschieht dies nun automatisch, sofern das Tangible nicht permanent benutzt wird. Das Schließen per Druckknopf funktioniert aber weiterhin.

Zur Erhöhung der Übersichtlichkeit wird jeder Tangible-Kategorie eine Farbe zugeordnet (beispielsweise sind alle Input-Tangibles mit der Farbe Gelb assoziiert). Wird nun ein Tangible als Input-Tangible deklariert, färben sich der gewählte Menüeintrag und der Bereich auf der Tabletop-Oberfläche, auf dem das entsprechende Tangible steht, in der entsprechenden Farbe. Somit ist für den Benutzer immer anhand der Farben ersichtlich, mit welchem Tangible er gerade arbeitet.

Es wurde eine Bestätigungsmeldung gewünscht, bevor der Nutzer einem Tangible eine andere Kategorie zuweist und dadurch die Einstellungen, die er im Untermenü gemacht hat, überschreiben würde. Diese Meldung wird ebenfalls im finalen Konzept enthalten sein.

Ein großer Kritikpunkt vieler Studienteilnehmer war der Bildschirm des Tangibles, der zu wenige und teilweise nicht aktuelle Informationen anzeigt. Im finalen Konzept wird bei jeder Änderung, die der Nutzer über das virtuelle Menü tätigt, das betreffende Tangible sofort aktualisiert. Der Bildschirm wird somit stets korrekte Informationen anzeigen.

Befindet sich das Tangible in einer Kette mit anderen Tangibles, so werden die Änderungen gegebenenfalls auch von weiteren Tangibles in der Kette angenommen.

Alle anderen Punkte, die in Abschnitt 6.2.2 beschrieben sind, werden nicht in den Prototyp übernommen, da sie mit bereits eingeplanten Interaktionsschritten kollidieren (etwa das Verbinden von Tangibles und das Kopieren von Tangible-Einstellungen), nur schwer umsetzbar sind (Text-to-Speech einer Audioaufnahme mit dem Notiz-Tangible) oder nicht sinnvoll sind (etwa das Erweitern des virtuellen Menüs um eine Lösch- und Notizfunktion; die Tabletop-Oberfläche ist begrenzt).

6.3 Interaktionskonzept des Prototyps

Dieses Kapitel beschreibt das Interaktionskonzept, welches dem Prototyp zugrunde liegt. Es basiert auf den Anforderungen (siehe Kapitel 5), den Vorüberlegungen (siehe Abschnitt 6.1) und den Ergebnissen der Paper-Prototyping-Studie (siehe Abschnitt 6.2.2). Im Folgenden werden nun die einzelnen Bereiche des Konzepts genau beschrieben, wie sie im Prototyp umgesetzt sind.

6.3.1 Typen von Tangibles

Es gibt zwei verschiedene Typen von Tangibles, die bereits in den Vorüberlegungen (siehe Abschnitt 6.1) vorhanden waren. Der erste Typ ist generisch und bekommt seine Funktion zur Laufzeit über ein virtuelles Menü zugewiesen. Er verfügt über folgende Komponenten:

- Bluetooth-Modul zur Kommunikation
- Druckknopf zum Öffnen und Schließen des virtuellen Menüs
- Schalter zum Ein- und Ausschalten des Tangibles
- USB Stromversorgung
- Batterie
- Mikroprozessor zum Implementieren und Ausführen der Programmlogik des Tangibles
- Beschleunigungssensor zur Gestenerkennung
- Bildschirm zur Informationsausgabe

Aufgrund der Ausstattung mit einem Prozessor, Sensoren und Kommunikationsmöglichkeit wird dieser Tangible-Typ als *active Tangible* bezeichnet. Von diesem Typ müssen mehrere Einheiten zur Verfügung stehen, sodass sie verkettet werden und ihr Potenzial ausspielen können. Eine detaillierte Beschreibung ist in Abschnitt 7.1 zu finden. Der zweite Typ ist ein Information-Tangible und dient dazu, Metadaten der den Tangibles zugewiesenen Daten oder Visualisierungen auf externen Geräten, die ins System eingebunden sind, anzuzeigen. Dieser Typ ist passiv (*Passive Tangible*) und benötigt lediglich einen Microsoft Byte Tag an der Unterseite. Hierfür können zahlreiche Gegenstände verwendet werden (etwa ein Plexiglaszylinder, ein würfelförmiger Karton oder eine Tasse).

6.3.2 Startzustand des Interaktionskonzepts

Am Anfang ist die Tabletop-Oberfläche abgesehen von einem Hinweis, der den Nutzer auffordert, ein Tangible auf den Tisch zu stellen, leer und es stehen keine Tangibles darauf (siehe Abbildung 35). Die Active Tangibles haben sich per Bluetooth mit dem Tabletop-Computer verbunden und ihr Bildschirm zeigt diesen Status dem Nutzer an.

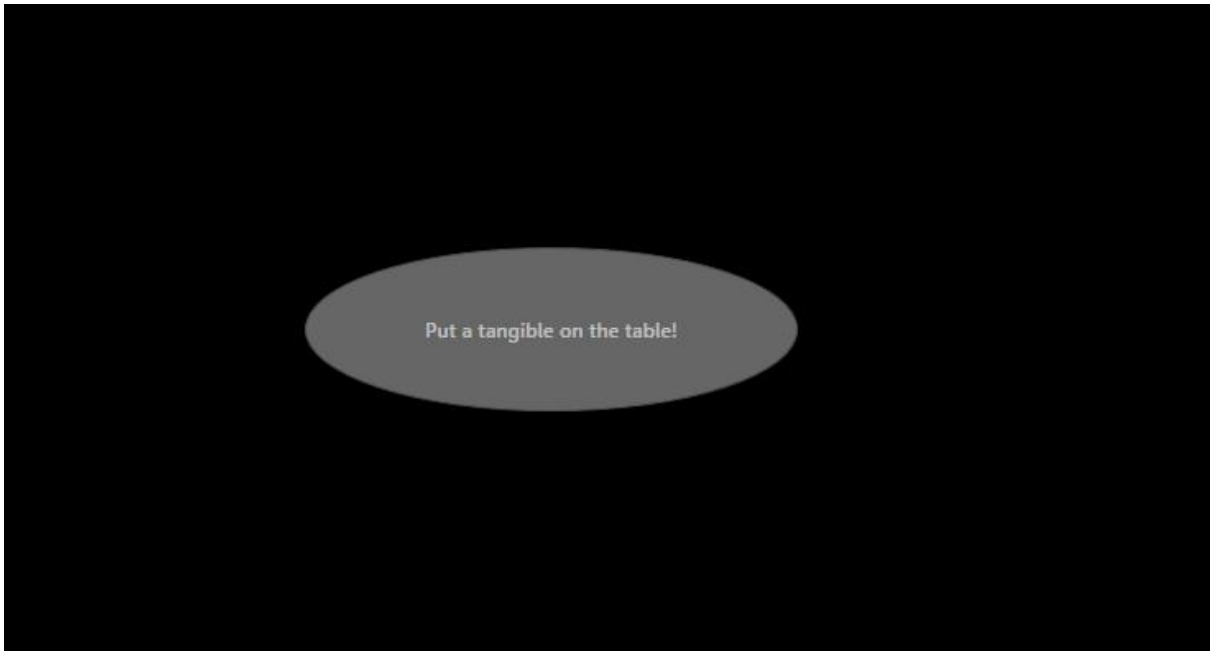


Abbildung 35 – Bild der Tabletop-Oberfläche zu Beginn. Ein Hinweissfeld leitet den Benutzer an, ein Tangible auf den Tisch zu stellen.

6.3.3 Erstes Tangible wird platziert

Der Nutzer platziert ein Active Tangible auf dem Tisch, worauf die angezeigte Hilfemeldung verschwindet. Es erscheint ein virtuelles Menü oberhalb des platzierten Tangibles (siehe Abbildung 36), in dem der Nutzer die Funktion festlegen kann. Es stehen die folgenden Möglichkeiten zur Auswahl:

- *Data Source*
- *Visualize*
- *Filter*
- *Highlight*

Mittels *Data Source* können Daten zur Analyse in das System eingegeben werden und über *Visualize* lassen sie sich in einer Visualisierungsform ausgeben. Die Punkte *Filter* und *Highlight* sind Modifikatoren und dienen zum Verändern einer Eingabe. Mit Filter-Tangibles lassen sich Datenmengen eingrenzen und mit Highlight-Tangibles können Daten in Visualisierungen hervorgehoben werden. Wählt der Nutzer einen Eintrag aus, wird dieser farblich hinterlegt und es erscheint ein Untermenü, in welchem Parameter ausgewählt werden können. Das Untermenü des Eintrags *Data Source* listet alle Datensätze auf, die zur Analyse zur Verfügung stehen. Der Nutzer kann nun über Checkboxes diejenigen Datensätze auswählen, die er analysieren möchte. Die Untermenüs für die Punkte *Filter* und *Highlight* sehen genauso aus wie das Untermenü *Data Source*. Im Punkt *Visualize* sind im Prototyp zwei Visualisierungsformen implementiert: Heatmaps und Scan-Path-Diagramme. Jedoch können hier prinzipiell weitere Arten von Visualisierungen eingebaut werden.

Menü und Untermenü sind frei verschiebbar auf der Tabletop-Oberfläche, werden jedoch immer oberhalb des zugehörigen Tangibles platziert, wenn dieses verschoben wird. Werden die Menüs einige Sekunden lang nicht benutzt, blenden sie sich automatisch aus. Alternativ kann der Nutzer dies auch explizit über das Drücken eines virtuellen Buttons im Untermenü oder mit dem seitlichen Druckknopf am Tangible erreichen.

Der Bildschirm des Tangibles zeigt immer aktuelle Informationen über die zugewiesene Funktion an. Bei einer Änderung wird der Bildschirm automatisch aktualisiert.

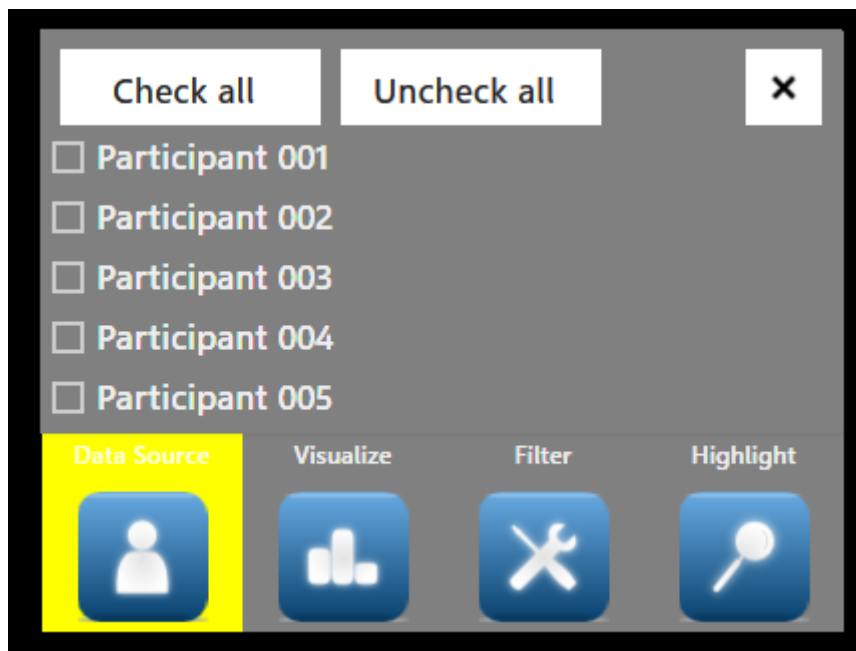


Abbildung 36 – Bild des virtuellen Menüs eines Tangibles. Es wird oberhalb des platzierten Tangibles auf der Tabletop-Oberfläche dargestellt und kann verschoben werden. Der Benutzer kann aus vier Funktionen auswählen und im Untermenü weitere Einstellungen tätigen. Die Farbe des gewählten Menüeintrags wird ebenfalls als virtuelles animiertes Rechteck um das Tangible herum auf der Tabletop-Oberfläche dargestellt. Jede Tangible-Kategorie besitzt eine eigene Farbe.

6.3.4 Ändern und Löschen von Zuweisungen

Der Nutzer hat ein Tangible auf dem Tisch platziert, die Funktion *Data Source* ausgewählt, im Untermenü mehrere Datensätze ausgewählt und die Menüs ausgeblendet. Um die Zuweisung zu ändern, betätigt er den seitlichen Druckknopf am Tangible. Dadurch öffnen sich das Auswahlmenü sowie das Untermenü und die Datenzuweisung kann verändert werden. Wählt der Nutzer eine andere Funktion (etwa *Visualize*), erscheint eine Bestätigungsmeldung mit dem Hinweis, dass die Zuweisungen der Datensätze gelöscht werden, wenn zuvor bereits eine andere Funktion bei diesem Tangible gewählt. Bestätigt er diesen Dialog nicht, wird auch die Funktion nicht geändert. Um ein Tangible in den Ausgangszustand zurückzusetzen, muss es lediglich kurz kräftig geschüttelt werden, sodass alle Daten- und Funktionszuweisungen gelöscht werden. Zuweisungen bleiben erhalten, wenn das Tangible vom Tisch entfernt wird.

6.3.5 Farben

Jede Tangible-Kategorie besitzt ihre eigene Signalfarbe. Standardmäßig steht gelb für Input-Tangibles, grün für Modifier-Tangibles, rot für Output-Tangibles und weiß für Information-Tangibles. Ein kleiner Bereich der Tabletop-Oberfläche um das Tangible leuchtet immer in der passenden Farbe und die Menüelemente sind damit hinterlegt (siehe Abbildung 36 und 37). Somit weiß der Nutzer auch ohne, dass das Menü geöffnet ist oder er auf den Bildschirm des Tangibles schaut, mit welcher Tangible-Kategorie er gerade arbeitet.

6.3.6 Verkettung mehrerer Tangibles

Zu jedem Tangible werden die Umriss eines weiteren Tangibles auf der Tabletop-Oberfläche dargestellt. Wird in diesem Bereich ein Tangible platziert, verbinden sich die

beiden, wenn ihre Kategorien kompatibel sind. Die Kategorie eines Tangibles richtet sich nach dessen zugewiesener Funktion: *Data Source* repräsentiert ein Input-Tangible, *Visualize* ein Output-Tangible und *Filter* und *Highlight* resultieren in einem Modifier-Tangible. Die Verbindungsregeln sind in Tabelle 2, Kapitel 6.1 dargestellt.

Eine Verbindung wird graphisch als animierter Verbindungsbalken angezeigt. Dabei wandert die Animation immer aus Richtung von Input-Tangibles in Richtung von Output-Tangibles und repräsentiert den Datenfluss. Kann die Datenflussrichtung nicht bestimmt werden (etwa wenn nur zwei Filter-Tangibles miteinander verbunden werden), so wird beim Hinzufügen eines anderen Tangibles zur Kette die Datenflussrichtung neu berechnet. Spätestens wenn ein Input oder Output-Tangible in der Kette vorhanden ist, kann die Datenflussrichtung bestimmt werden.

Wird ein Tangible auf dem Tabletop-Computer verschoben, färben sich die Umrisse der übrigen Tangibles entweder rot, wenn sie inkompatibel sind, oder grün, wenn sie verbunden werden können. So kann schnell überprüft werden, ob zwei Tangibles verbunden werden können (siehe Abbildung 37).

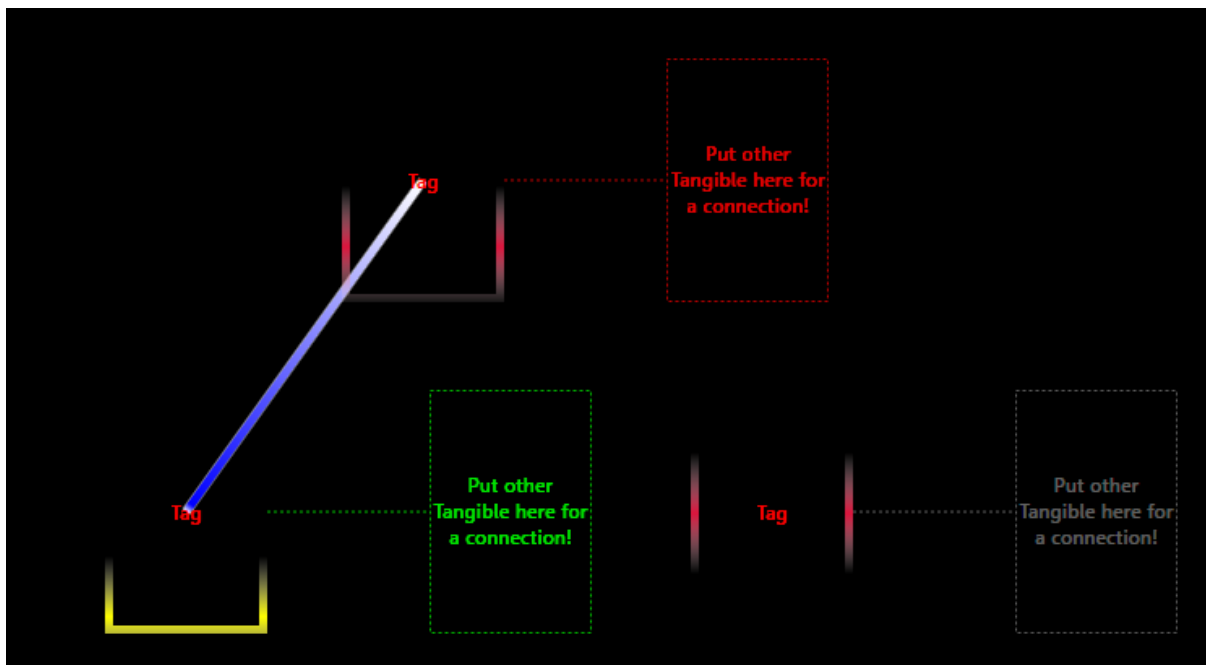


Abbildung 37 – Darstellung der Tangible-Verkettung auf der Tabletop-Oberfläche. Es sind drei Tangibles platziert, ein Input-Tangible (gelb) und zwei Output-Tangibles (rot). Ein Input und ein Output-Tangible sind verbunden, die Animation läuft vom Input zum Output. Bei Bewegungen des nicht verbundenen Tangibles werden die Umrisse der anderen beiden Tangibles eingefärbt. Eine grüne Färbung signalisiert, dass eine Verbindung möglich ist. Bei einer roten Färbung ist dies nicht der Fall.

6.3.7 Visualisierungen anzeigen

Um Daten in einer Visualisierung anzeigen zu können, ist mindestens eine Kette von zwei Tangibles (ein Input und ein Output-Tangible) erforderlich, die verbunden sind. Im Input-Tangible muss mindestens ein Datensatz ausgewählt sein, der angezeigt werden kann. Der Nutzer hat bei der Visualisierung die Wahl zwischen einer Heatmap-Darstellung und einem Scan-Path-Diagramm. Beide können im Untermenü des Menüpunktes *Visualize* ausgewählt werden. Während der Generierung der Visualisierung wird ein Ladeindikator angezeigt, der wieder verschwindet, sobald die Visualisierung angezeigt wird. Die Visualisierung zeigt sämtliche Datensätze an, die durch die Tangible-Kette von Input-Tangibles weitergeleitet werden. Ändert sich die Tangible-Kette, ändert sich gegebenenfalls auch die Visualisierung.

Visualisierungen können gedreht, skaliert und verschoben werden. Bei vielen Datensätzen sollte jedoch leistungsstarke Hardware zur Verfügung stehen.

6.3.8 Filtern und Hervorheben

Um Datensätze einzugrenzen, können Filter-Tangibles (kurz *Filter*) eingesetzt werden. Wird ein Filter (Modifier-Tangible) mit einem Input-Tangible verbunden, leitet der Filter die Schnittmenge der Datensätze der beiden Tangibles weiter. Beispielsweise sind im Input-Tangible die Datensätze 1-5 aktiviert und im Filter die Datensätze 2,4 und 6. Wird nun ein Output-Tangible an den Filter gekettet, leitet dieser die Datensätze 2 und 4 an den Filter weiter (siehe Abbildung 38). Prinzipiell kann die Filterfunktionalität auch einfach dadurch erreicht werden, dass im Input-Tangible die Datensätze angepasst werden. Durch Austauschen von Filtern entfällt jedoch das ständige Wechseln der Datensätze im Input-Tangible.

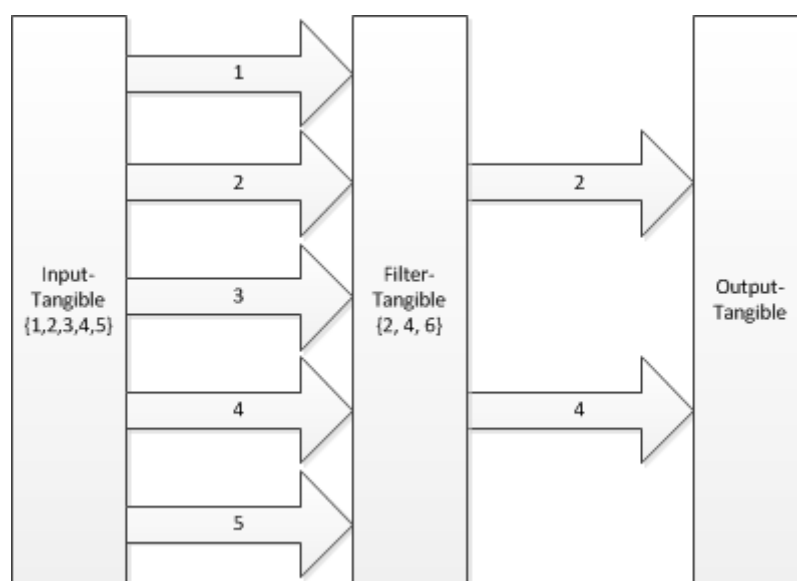


Abbildung 38 – Skizze der Funktion eines Filter-Tangibles. Die Eingangsmenge des Input-Tangibles umfasst fünf Elemente und wird durch den Filter auf zwei Elemente eingeschränkt, die zum Output-Tangible weitergeleitet werden.

Ein Highlight-Tangible (kurz *Highlighter*) dient zum Hervorheben von Datensätzen in Visualisierungen. Im Prototyp unterstützt dies nur das Scan-Path-Diagramm, da es in einer Heatmap keine sinnvolle Form der Hervorhebung einzelner Datensätze gibt. Wird ein Highlighter (Modifier) mit einem Input-Tangible verbunden, leitet er zwei Mengen weiter: Die Menge an Datensätzen, die hervorgehoben werden soll und die Menge an Datensätzen, die nicht hervorgehoben werden soll. Sind beispielsweise im Input-Tangible die Datensätze 1-5 aktiviert und im Highlighter die Datensätze 1 und 2, dann zeigt eine mit dem Highlighter verbundene Scan-Path-Visualisierung die Datensätze 1 und 2 in Signalfarben und die Datensätze 3,4 und 5 in einer Einheitsfarbe mit erhöhter Transparenz (siehe Abbildung 39).

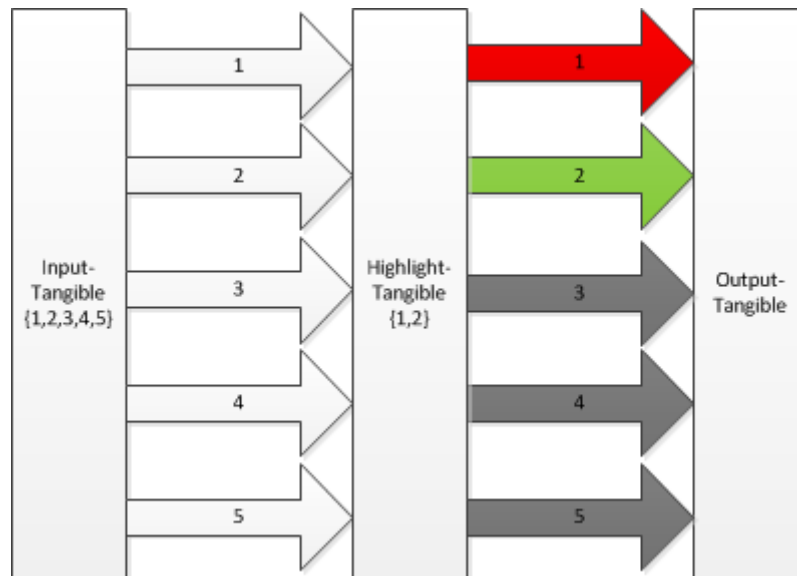


Abbildung 39 - Skizze der Funktion eines Highlight-Tangibles. Die Eingangsmenge des Input-Tangibles umfasst fünf Elemente und wird durch das Highlight-Tangible in zwei Mengen aufgeteilt: Die erste Menge (Element 1 und 2) wird farblich hervorgehoben, die zweite Menge (Element 3,4 und 5) wird in einer Einheitsfarbe dargestellt. Beide Mengen werden zum Output-Tangible weitergeleitet.

So lassen sich auch aus einer größeren Menge an Datensätzen noch Informationen herauslesen (siehe Abbildung 40). Ähnlich wie beim Filter kann auch die Funktion des Highlighters prinzipiell durch Ausschalten einiger Datensätze im Input-Tangible realisiert werden.

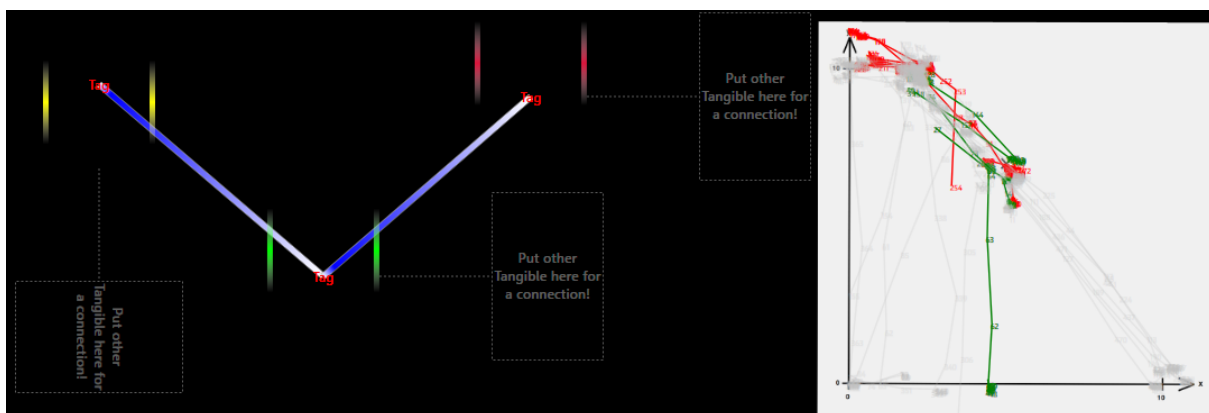


Abbildung 40 – Auf der Tabletop-Oberfläche sind ein Input-Tangible (gelb), ein Highlight-Tangible (grün) und ein Output-Tangible (rot) miteinander verbunden. Im Input-Tangible sind vier Datensätze aktiviert, im Highlight-Tangible sind zwei Datensätze aktiviert. Das Scan-Path-Diagramm zeigt vier Datensätze, wovon zwei in Signalfarben (rot und grün) und zwei in einer Einheitsfarbe (grau) dargestellt werden.

6.3.9 Externen Bildschirm einbeziehen

Um externe Bildschirme einzubinden, wird das Information-Tangible benutzt. Es unterliegt keinen Verbindungsregeln und kann mit allen Kategorien von Tangibles verbunden werden. Die Daten fließen immer zum Information-Tangible. Dieses Tangible besitzt kein Einstellungsmenü. Sobald es mit einem Tangible verbunden ist, werden die Metainformationen der Datensätze (bei Input- und Modifier-Tangibles) beziehungsweise die dargestellte Visualisierung (bei Output-Tangibles) auf einen externen Monitor verschoben. Wird das Tangible wieder entfernt, verschwinden die Informationen wieder und die Visualisierung wird wieder auf den Tabletop-Computer verschoben.

7 Prototyp

In diesem Kapitel wird der entwickelte Prototyp vorgestellt, der wesentliche Punkte des Interaktionskonzeptes umsetzt. So demonstriert er beispielsweise das Verbinden von Tangibles, den Datenaustausch zwischen ihnen, die Informationsdarstellung auf dem jeweiligen Bildschirm eines Tangibles, die Erweiterung der Oberfläche um externe Geräte und die Unterstützung von Gesten mithilfe der Tangibles. Anhand dieser Kernelemente kann das Gesamtkonzept gut verdeutlicht und nachvollzogen werden.

Kapitel 7.1 befasst sich mit der Entwicklung der Tangibles. Darin werden unter anderem die Punkte Funktionsweise der Tangibles, Materialkosten, Gehäuseproduktion und Softwareprogrammierung beschrieben. Kapitel 7.2 beschreibt die Software (Benutzerschnittstelle, Kommunikation, Graphstruktur, Datenhaltung), die auf dem Tabletop-Computer läuft.

7.1 Tangibles

Um ein Tangible User Interface zu realisieren, sind Tangibles unerlässlich. Diese wurden im Rahmen dieser Arbeit mittels eines Rapid-Prototyping-Verfahrens selbst entwickelt. Kapitel 7.1.1 beschreibt die Funktionsweise, Kapitel 7.1.2 die Entwicklung und Kapitel 7.1.3 die Software der Tangibles.

7.1.1 Funktionsweise

Die Kommunikation zwischen Tangibles und Tabletop-Computer läuft über eine bidirektionale drahtlose Verbindung, wobei der Tabletop-Computer Tangibles zusätzlich an ihren Byte Tags [14] erkennen kann, welche sich an der Unterseite der Tangibles befinden. Der Bildschirm eines Tangibles zeigt Hinweise zur aktuellen Funktionalität an, die im Zusammenhang mit der Darstellung auf der Tabletop-Oberfläche dafür sorgen, dass der Nutzer immer über den Zweck des Tangibles Bescheid weiß.

7.1.2 Entwicklung

Im Folgenden wird kurz auf den Entwicklungsprozess der Tangibles eingegangen. Dabei geht es um die Prototyping-Technik, aufgetretene Probleme und Maßnahmen zu deren Lösung.

Version 1 (Codename „Spinne“)

Eine erste Version wurde mit dem .NET Gadgeteer Kit entwickelt. Die benötigten Teile werden einfach auf einem Mainboard zusammensteckt und anschließend mit der objektorientierten Sprache C# und dem .NET Micro Framework programmiert. Tabelle 6 zeigt eine Auflistung aller Komponenten, die im ersten Tangible-Prototyp enthalten sind.

Komponente	Beschreibung	Preis (in Euro)
FEZ Spider Mainboard	Andockpunkt für Module	88,65
Bluetooth Module	Bluetooth-Kommunikation	29,53
LED 7C Modul	Mehrfarbige LED	5,14
Button Modul	Druckknopf	3,66
Accelerometer Modul	Beschleunigungssensor	9,53
UC Battery 4xAA Modul	Stromversorgung	14,75
OLED Display Modul	Bildschirm	<u>10,72</u>
		161,98

Tabelle 6 – Auflistung aller Komponenten des ersten Tangible-Prototyps auf .NET Gadgeteer-Basis. Der Gesamtpreis von knapp 162 Euro beinhaltet keine Kabel-, Montage- oder Gehäuseproduktionskosten. Alle Komponenten sind frei im Handel erhältlich^{9,10}

Nach der erfolgreichen Testimplementierung der benötigten Funktionen wurde ein Gehäuse erstellt. Eine zu der Zeit aktuelle Arbeit am Institut für Visualisierung und Interaktive Systeme der Universität Stuttgart beschäftigte sich mit der einfachen Modellierung von Gehäusevorlagen ohne CAD-Kenntnisse speziell für Module des .NET Gadgeteer Kits [33]. Mit dem daraus erstellten Modell konnte innerhalb weniger Minuten aus einer Holzplatte mit einem Lasercutter eine Schale produziert werden, in der alle Komponenten untergebracht werden konnten (siehe Abbildung 41).

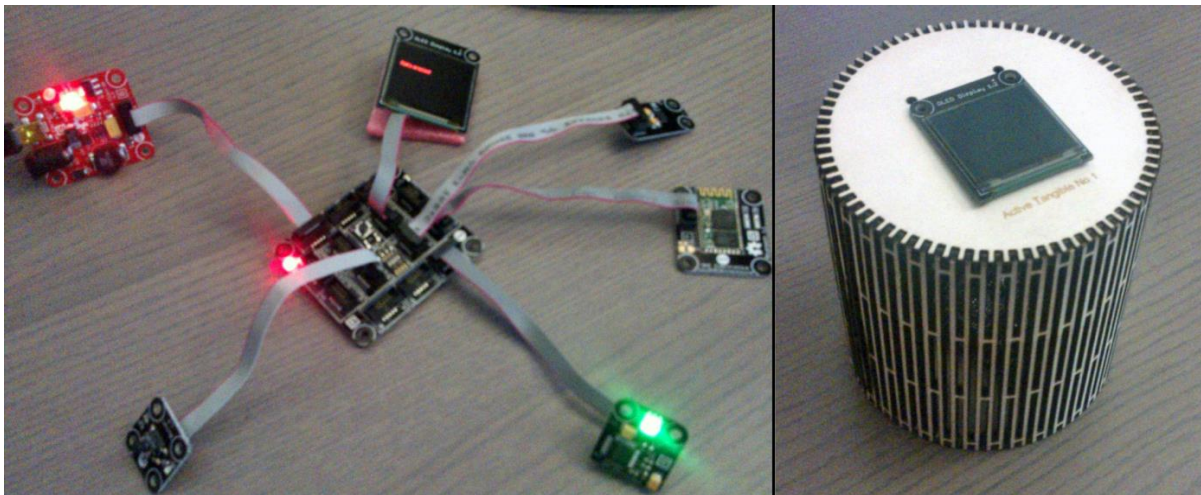


Abbildung 41 – Im linken Bild ist die Darstellung der verbundenen .NET Gadgeteer Module ohne Gehäuse („Spinne“) zu sehen. Im rechten Bild ist das aus einer Holzplatte lasergeschnittene Gehäuse mit den Komponenten darin zu sehen.

Der Umfang der Tangible-Hülle orientiert sich an der Komponente mit der größten Grundfläche. Die Abmessungen des Mainboards sind 5,7 cm in der Breite und 5,2 cm in der Länge, sodass mit der Dicke der Holzplatte (0,4 cm) und einem Sicherheitsabstand bei einem runden Gehäuse ein Durchmesser von 10cm anfällt. Wenn nun beispielsweise drei Tangibles auf dem Tabletop stehen und zusätzlich noch zwei Visualisierungen angezeigt werden, ist der Platz nahezu aufgebraucht. Auch eine vertikale Positionierung des Mainboards brachte keine Verbesserung, da die Module nicht mehr darauf gestapelt werden konnten, sondern seitlich montiert werden mussten. Im Endeffekt ergab sich so ein sogar noch größerer Umfang.

⁹ <https://www.ghielectronics.com/catalog/category/517>

¹⁰ <http://www.seeedstudio.com/depot/>

Version 2 (Codename „Block“)

Nachdem der erste Prototyp durch die zu großen Abmessungen nicht verwendet werden konnte, wurde ein zweiter Prototyp auf Basis der Arduino-Plattform entwickelt. Im Vergleich zum .NET Gadgeteer Kit lassen sich mit der Arduino-Plattform kompaktere Formen realisieren, da es wesentlich kleinere Mainboards gibt. Allerdings benötigt dieser Ansatz elektrotechnische Kenntnisse, um die Module korrekt anzuschließen. Eine falsche Verkabelung kann bis zur Zerstörung eines Moduls führen.

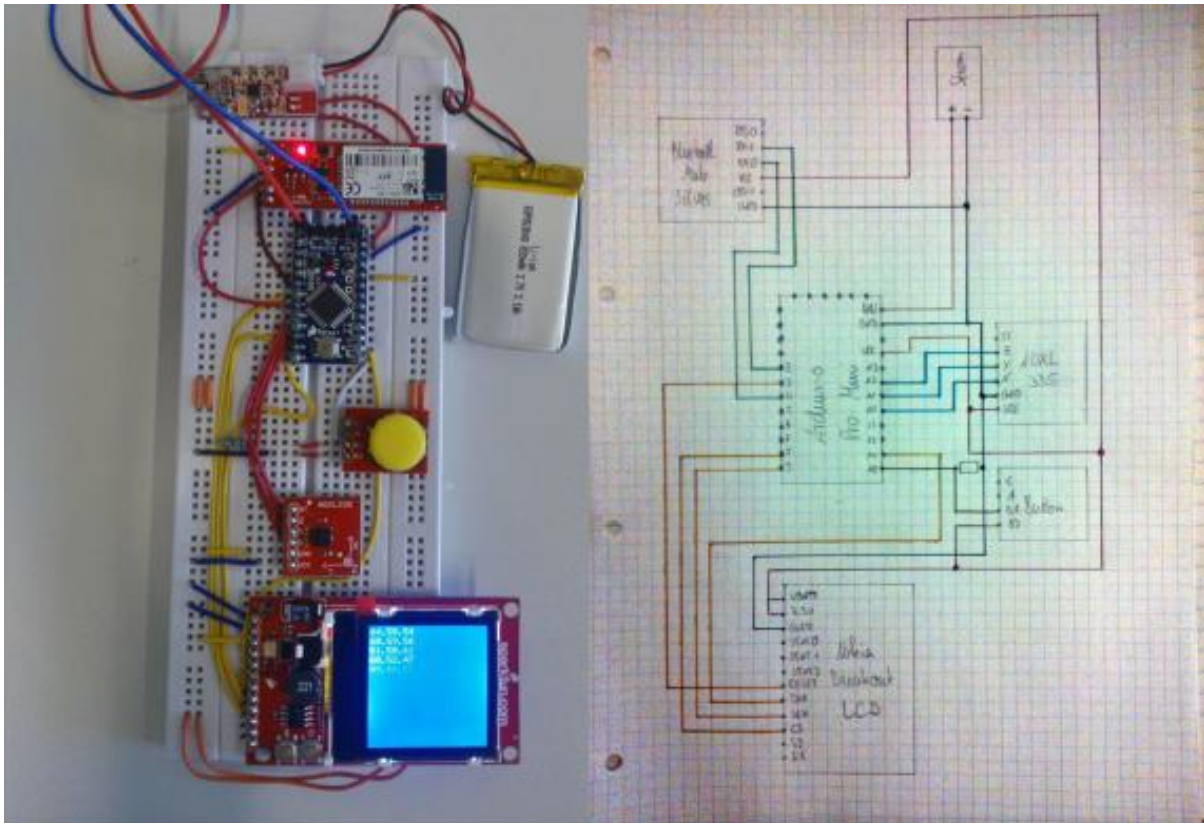


Abbildung 42 – Fertig montiertes arduino-basiertes Tangible auf einem Steckbrett (links). Die Batterie (rot-schwarzes Kabel) ist mit einem Klettverschluss an der Hinterseite des Steckbretts befestigt. Ein eigens per Lasercutting hergestellter Standfuß mit einem Microsoft Byte Tag an der Unterseite erlaubt bereits die Nutzung als vollwertiges Interaktionselement auf dem PixelSense Tabletop-Computer. Auf der rechten Seite ist ein Schaltplan zu sehen, um die Verkabelung des Tangibles nachvollziehen zu können.

Die einzelnen Komponenten wurden zunächst auf einem Steckbrett verkabelt (siehe Abbildung 42) und anschließend die Software implementiert. Für die Programmierung wurde die Arduino Entwicklungsumgebung mit der Sprache C benutzt. Im weiteren Verlauf wurden die Komponenten platzsparend mit Löten und Kleben übereinander auf einer Platine befestigt. Abschließend wurde ein Gehäuse digital modelliert und per 3D-Druck (FDM-Verfahren) hergestellt. Bei der Konstruktion der Tangibles wurde darauf geachtet, dass alle Teile nicht permanent miteinander verbunden sind, sondern bei Bedarf (etwa ein Defekt) mit wenig Aufwand ersetzt werden können.

Komponente	Beschreibung	Preis (in Euro)
Arduino Pro Mini 328 (3.3V/8MHz)	Mikroprozessor	7,35
ADXL335 Accelerometer	Beschleunigungssensor	18,44
Color LCD Breakout	Bildschirm	25,83
LiPo Charger Basic Micro-USB	Stromversorgung	5,88
Polymer Lithium Ion Battery 850 mAh	Stromversorgung	6,61
Bluetooth Mate Silver	Bluetooth-Kommunikation	29,53
Mini Push Button Switch	Druckknopf	0,26
Slide Switch	Schalter	<u>0,55</u>
		94,45

Tabelle 7 - Auflistung aller Komponenten des zweiten Tangible-Prototyps auf Arduino-Basis. Der Gesamtpreis von knapp 95 Euro beinhaltet keine Kabel-, Montage- oder Gehäuseproduktionskosten. Alle Komponenten sind frei im Handel erhältlich¹¹

Das fertige Tangible im Gehäuse ist knapp 10 cm breit, etwa 15 cm lang und besitzt eine Höhe von ungefähr 9,5 cm (siehe Abbildung 43 und 44). Der Microsoft Byte Tag ist an der Unterseite aufgeklebt, sodass das Tangible in liegender Position verwendet werden kann. Wenn sich der Einschalter und der Druckknopf aus Sicht des Benutzers rechts und der Strom- sowie Datenanschluss links befinden, ist das Tangible korrekt ausgerichtet. Eine Nutzung in vertikaler Form ist prinzipiell auch möglich, jedoch aufgrund der Konstruktion des Tangibles nicht sinnvoll, da der Informationsbildschirm in dem Fall nur umständlich ablesbar ist.

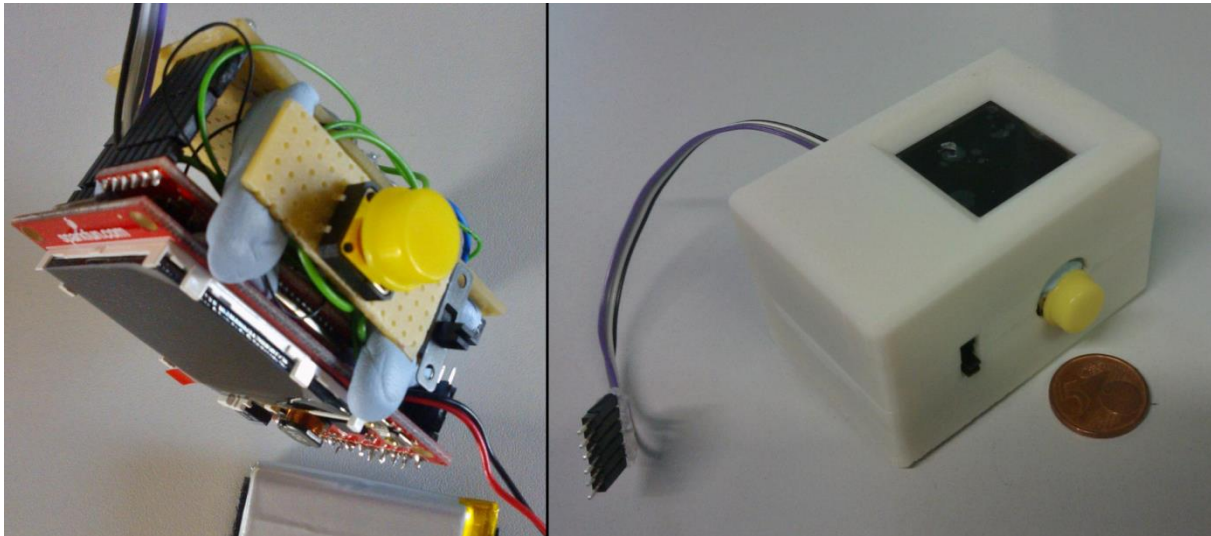


Abbildung 43 – Im linken Bild ist ein fertig gebautes Tangible ohne Gehäuse zu sehen, im rechten Bild ist dasselbe Tangible in einem per 3D-Druck hergestellten Gehäuse zu sehen („Block“). Das Gehäuse ist etwa 10 cm breit, knapp 15 cm lang und ungefähr 9,5 cm hoch.

¹¹ <https://www.sparkfun.com/>

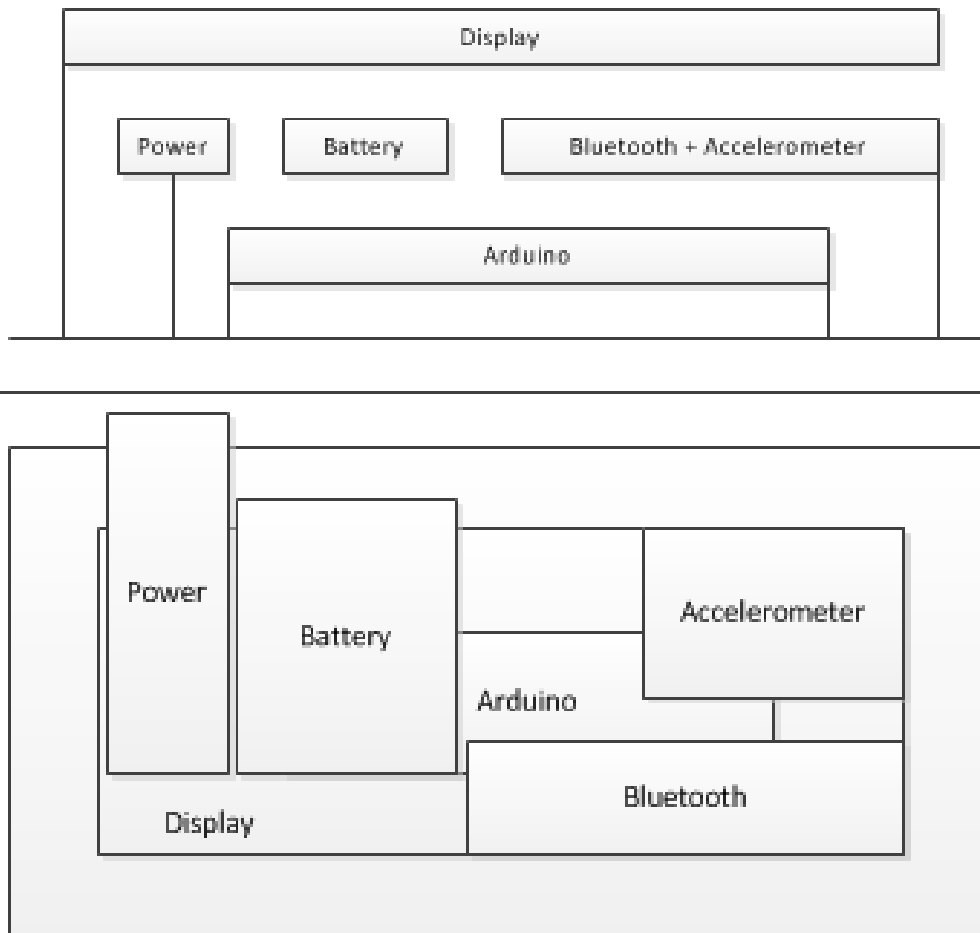


Abbildung 44 – Im oberen Bild ist die Seitenansicht eines konstruierten Tangibles zur Verwendung im entwickelten Interaktionskonzept zu sehen. Die Platzierung der Komponenten Bildschirm, USB Stromversorgung, Batterie, Mikroprozessor, Bluetooth-Modul und Beschleunigungssensor wird durch BluTack [34] fixiert. Im unteren Bild ist die Draufsicht zu sehen.

Das Tangible benötigt im Betrieb durchschnittlich 154mA Strom, wenn alle Komponenten aktiv sind. Bei Verwendung einer 850mAh Batterie ergibt sich somit eine Laufzeit von knapp 5,5 Stunden, bevor die Batterie geladen werden muss. Der größte Verbraucher aller eingesetzten Komponenten ist der Bildschirm. Ohne diesen wird der Verbrauch um beinahe 90% auf lediglich 17 mA im Mittel reduziert. Zur Realisierung von Stromsparmaßnahmen ist es daher sinnvoll, den Bildschirm abzuschalten, wenn möglich. Im Prototyp sind keine Stromsparmaßnahmen implementiert.

7.1.3 Tangible-Software

Die Software für die Tangibles umfasst das Senden und Empfangen von Bluetooth-Nachrichten, das Erkennen einer Schüttelgeste und das Ausgeben von Informationen auf dem Bildschirm.

Eine Bluetooth-Verbindung kann über eine serielle Schnittstelle realisiert und mit der in der Arduino-Entwicklungsumgebung integrierten SoftwareSerial-Bibliothek¹² einfach verwendet werden, indem die Befehle `read()` zum Lesen von Daten aus dem Bluetooth-Datenstrom und `print()` zum Schreiben in den Datenstrom benutzt werden. Zur Verarbeitung von eingehenden Befehlen wurde ein Interpreter implementiert (siehe Abbildung 45), der

¹² <http://arduino.cc/de/Reference/SoftwareSerial>

Befehle erkennen kann, die ein bestimmtes Format aufweisen. Das Befehlsformat sieht folgendermaßen aus:

- \$ + BluetoothAdresse des Senders + < + Befehl + & (Tangible zum Tabletop)
- \$ + > + Befehl + & (Tabletop zu Tangible)

Mögliche zu sendende Befehle vom Tabletop sind:

- DISPLAYCLEAR (löscht den Bildschirminhalt des Zieltangibles)
- DISPLAYON (aktiviert den Bildschirm des Zieltangibles)
- DISPLAYOFF (deaktiviert den Bildschirm des Zieltangibles)
- PING (sendet eine Ping-Nachricht an das Zieltangible)
- TEXTIN>Überschrift: Inhalt (stellt Text im Bildschirm des Zieltangibles dar)

Mögliche zu sendende Befehle vom Tangible sind:

- MENUTRIGGER (zeigt oder versteckt das Menü des sendenden Tangibles)
- DATACLEAR (setzt das sendende Tangible in den Ausgangszustand zurück)

```

////////////////////////////////////
// checks for a valid command frame
////////////////////////////////////
void commandInterpreter(){
  //Serial.println(command);
  if (command.equals("DISPLAYON")){
    GLCD.displayOn();
    bluetooth.println("$" + deviceAddress + "<ACK&");
    return;
  }
  if (command.equals("DISPLAYOFF")){
    GLCD.displayOff();
    bluetooth.println("$" + deviceAddress + "<ACK&");
    return;
  }
  if(command.equals("DISPLAYCLEAR")){
    clearDisplay();
    bluetooth.println("$" + deviceAddress + "<ACK&");
    return;
  }
  if (command.startsWith("TEXTIN>")){
    String header = command.substring(7, command.indexOf(':'));
    String content = command.substring(command.indexOf(':') + 1);
    printInformationMessage(header, content, 0, 0, 0);
    bluetooth.println("$" + deviceAddress + "<ACK&");
    return;
  }
  if (command.startsWith("PING")){
    bluetooth.println("$" + deviceAddress + "<ACK&");
    if (!connection){
      connection = true;
      printStatusMessage("connection established", 0, 255, 0);
    }
    return;
  }
  bluetooth.println("$" + deviceAddress + "<ERR&");
  return;
}

```

Abbildung 45 – Prozedur zum Erkennen von Befehlen, die der Tabletop-Computer an ein Tangible sendet. Je nach erkanntem Befehl wird eine entsprechende Aktion ausgelöst und eine Bestätigungsmeldung an den Tabletop-Computer gesendet. Kann ein Befehl nicht erkannt werden, wird eine Fehlermeldung geschickt.

Der Algorithmus zur Schüttelerkennung (siehe Abbildung 46) misst permanent die Beschleunigung auf allen drei Achsen und reagiert, wenn sich die Werte auf mindestens 2 Achsen innerhalb kurzer Zeit stark ändern. Um versehentliches Auslösen zu verhindern wurde der Schwellwert auf das 1,5-fache der Erdbeschleunigungskraft gesetzt (1,5g). Ist diese Schwelle überschritten, wird die Nachricht DATACLEAR via Bluetooth an den Tabletop-Computer gesendet, der auf die Geste entsprechend reagieren kann.

```

/////////////////////////////////////////////////////////////////
// updates the accelerometer values
/////////////////////////////////////////////////////////////////
void updateAccelParams(float x, float y, float z){
    // ignore the first shake (values are initialized with 0)
    if (firstUpdate){
        lastX = x;
        lastY = y;
        lastZ = z;
        firstUpdate = false;
    }
    else {
        lastX = curX;
        lastY = curY;
        lastZ = curZ;
    }
    curX = x;
    curY = y;
    curZ = z;
}

/////////////////////////////////////////////////////////////////
// detects if the device has been shaken
/////////////////////////////////////////////////////////////////
boolean shakeDetected(){
    float deltaX = abs(lastX - curX);
    float deltaY = abs(lastY - curY);
    float deltaZ = abs(lastZ - curZ);
    return (deltaX > shakeThreshold && deltaY > shakeThreshold)
        || (deltaX > shakeThreshold && deltaZ > shakeThreshold)
        || (deltaY > shakeThreshold && deltaZ > shakeThreshold);
}

```

Abbildung 46 – Algorithmus zur Schüttelgestenerkennung des Tangibles. Es werden permanent die Beschleunigungswerte aller drei Achsen gemessen und mit den vorherigen Werten verglichen. Wird bei mindestens zwei Achsen der Unterschied zwischen Vorgänger- und aktuellem Wert größer als ein Schwellwert, ist eine Schüttelgeste ausgeführt worden.

Der Druckknopf am Tangible ist für das Anzeigen und Verstecken des Menüs zuständig. Wird er betätigt, wird die Nachricht MENUTRIGGER per Bluetooth an den Tabletop-Computer gesendet, der das Menü anzeigt, wenn es ausgeblendet ist und ausblendet, wenn es angezeigt ist.

Den Bildschirm kann man sich als Matrix vorstellen, wobei ein Zeiger immer auf eine bestimmte Position zeigt. Der Zeiger kann versetzt werden, um Text an einer anderen Position auszugeben. In der Standardeinstellung können knapp 21 Zeichen pro Zeile und 13 Zeilen pro Bildschirm angezeigt werden. Für kurze Informationstexte ist der Platz ausreichend. Zur Verdeutlichung kann im mehrfarbigen Display auch die Hintergrund- oder Textfarbe verändert werden.

7.2 Tabletop-Software

Als Tabletop-Computer kommt PixelSense von Microsoft (siehe Kapitel 2.3.2) zum Einsatz. Die Software dafür wird mit C# und dem .NET Framework von Microsoft in der Version 4.5 entwickelt. Zur Implementierung der Benutzeroberfläche kommt das *Windows Presentation Framework* (WPF) im Zusammenspiel mit dem *Surface Development Kit 2* (Surface SDK 2) zum Einsatz. Der Begriff Surface geht auf den früheren Namen des Tabletop-Computers zurück, bevor er in PixelSense umbenannt wurde. Die Anwendung läuft im Vollbildmodus und ist komplett durch Finger- und Tangible-Interaktion bedienbar, kann jedoch alternativ auch mit einer an den Tabletop angeschlossenen Maus bedient werden.

In den folgenden Abschnitten wird die implementierte Software beschrieben. Abschnitt 7.2.1 erläutert die für diese Arbeit relevanten Teile des zugrundeliegenden Datenmodells (siehe Abbildung 47). Abschnitt 7.2.2 gibt eine kurze Übersicht über die Architektur des Software-Systems. Abschnitt 7.2.3 beschreibt die Implementierung der Benutzerschnittstelle. Abschnitt 7.2.4 beschreibt den Ablauf und die Umsetzung der Kommunikation zwischen Tabletop und Tangibles. Abschnitt 7.2.5 beschreibt die Erzeugung der Graphstruktur aus verbundenen Tangibles und die Datenweiterleitung innerhalb der Struktur und Abschnitt 7.2.6 geht auf die Datenhaltung ein.

7.2.1 Modell der Eye-Tracking-Daten

Zur Veranschaulichung des im Rahmen dieser Arbeit entwickelten Interaktionskonzepts benötigt der Prototyp einen Datensatz. Hierfür kommen Daten zum Einsatz, die während Eye-Tracking-Studien aufgezeichnet wurden. Sie unterliegen dem Datenmodell, das in Abbildung 47 dargestellt ist. Es werden jedoch lediglich die Entitäten *Stimulus*, *Participant* und *Fixation* im Prototyp benutzt.

Ein Participant (Deutsch: Proband) ist ein Teilnehmer einer Eye-Tracking-Studie. Ein Stimulus ist eine Darstellung, die der Proband betrachtet, während seine Augenbewegungen erfasst werden und eine Fixation ist ein Punkt auf oder neben dem Stimulus, den der Teilnehmer für eine gewisse Zeit mit den Augen fixiert hat.

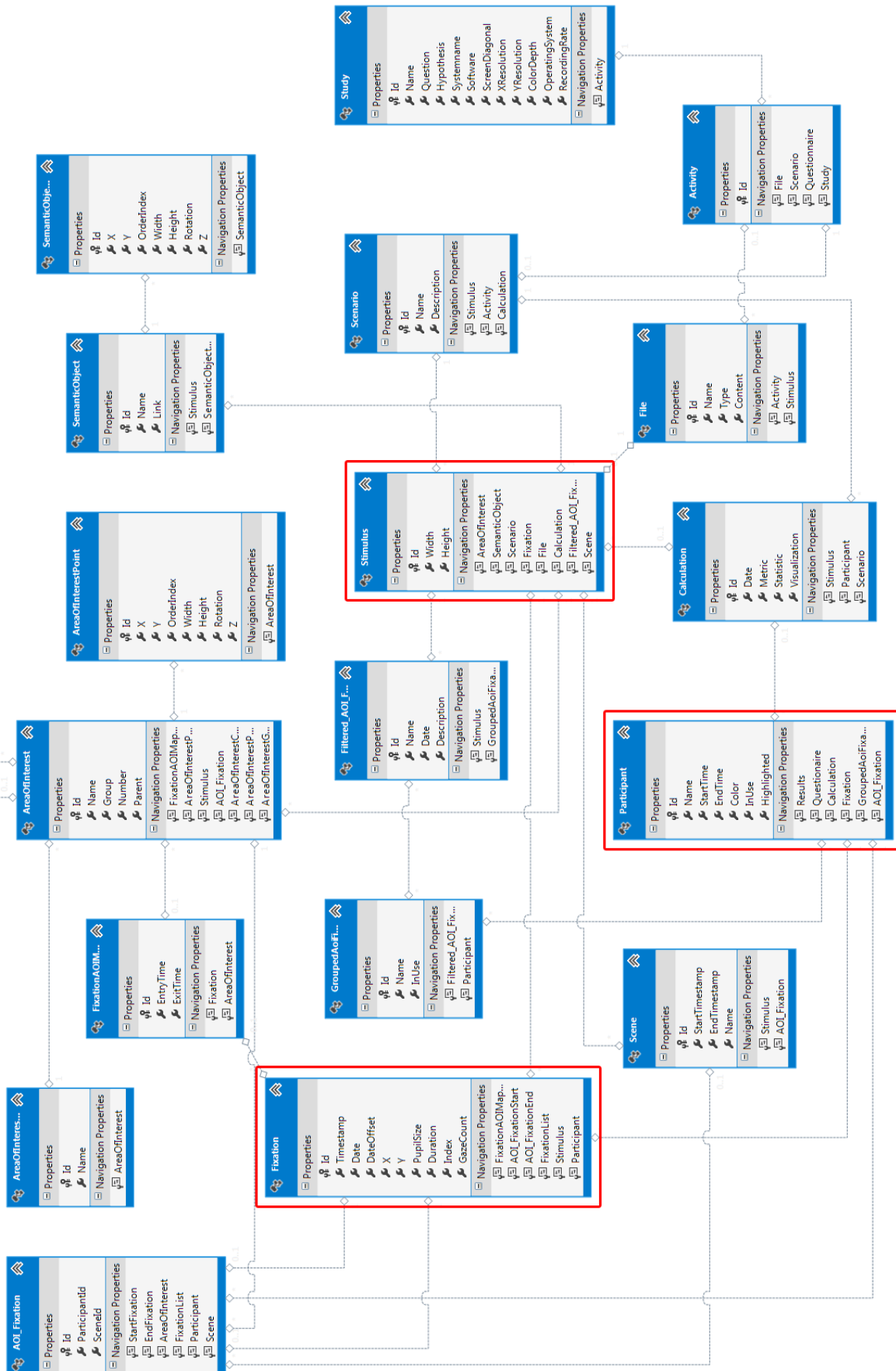


Abbildung 47 – Struktur der Eye-Tracking-Daten, die als Beispieldaten im Prototyp verwendet werden. Die tatsächlich verwendeten Entitäten sind Stimulus, Participant und Fixation, der Rest wird nicht aktiv genutzt.

7.2.2 Architekturüberblick

Die Architektur der Tabletop-Software gliedert sich in drei große Bereiche:

- Der erste Bereich umfasst alles, was die Benutzeroberfläche betrifft. Darunter fallen Datenvisualisierungen, Menüs, Verbindungslinien, Hilfsmeldungen, Ladezustandsindikatoren und Animationen.
- Der zweite große Bereich ist für die Kommunikation zwischen Tabletop und Tangible über Bluetooth verantwortlich. Anfallende Aufgaben sind das Verwalten aller aktiven Bluetooth-Geräte, das Senden, Empfangen und Interpretieren von Nachrichten und das Einleiten einer entsprechenden Aktion basierend auf dem Inhalt einer Nachricht.
- Der dritte große Bereich umfasst das Generieren von Graphen aus den verbundenen Tangibles. Für die Darstellung einer Visualisierung ist es nötig, dass sämtliche eingehenden Daten ermittelt werden können. Basierend auf dem Graph kann ermittelt werden, welche Daten zwischen welchen Tangibles weitergeleitet werden.

In den nächsten Abschnitten werden diese Bereiche nun detailliert beschrieben.

7.2.3 Benutzerschnittstelle

Dieser Abschnitt beschreibt die Benutzerschnittstelle und das Zusammenspiel der einzelnen Klassen inklusive der dafür relevanten Teile des Surface SDKs. Die Darstellung in Abbildung 48 illustriert den folgenden Text.

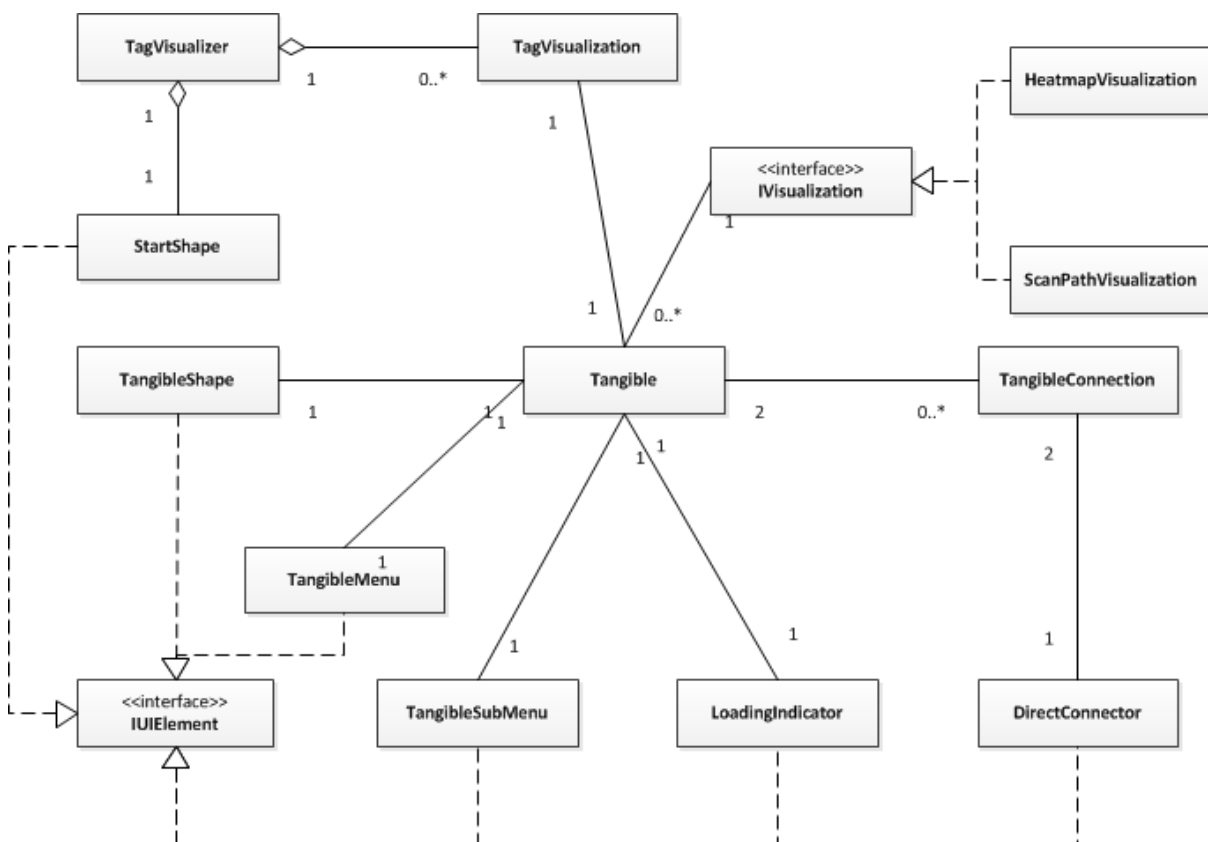


Abbildung 48 – Vereinfachtes Klassendiagramm bestehend aus allen Klassen, die Einfluss auf die Gestaltung der Benutzeroberfläche haben. Aus Gründen der Übersichtlichkeit werden die Methoden der Klassen nicht angezeigt.

Das Grundelement einer PixelSense Anwendung ist das *ScatterView* aus dem Surface SDK. Das ist ein Container, der beliebig viele *ScatterViewItems* aufnehmen kann. Alle in WPF

vorhandenen Elemente für graphische Oberfläche können in ein *ScatterViewItem* eingebettet werden. Somit können sie auf dem PixelSense verschoben, skaliert und rotiert werden. Alle Anzeigeelemente des Prototyps sind *ScatterViewItems*.

Zur Erkennung von Microsoft Byte Tags wird eine *TagVisualizer*-Komponente aus dem Surface SDK benötigt. Dieser kann mit den Tags interagieren und bietet Events für das Hinzufügen, Verschieben und Entfernen von Objekten an.

Zu Beginn ist nur das *StartShape*-Element sichtbar. Es verschwindet, sobald das erste Tangible auf der Oberfläche platziert wurde. Für jedes Tangible, das auf dem PixelSense platziert wird, wird ein *TagVisualization* Objekt im EventHandler des *TagVisualizers* erstellt. Das Tangible wird als animiertes Rechteck in der Farbe der gewählten Kategorie dargestellt (gelb für Input, grün für Modifier, rot für Output und transparent, wenn keine Kategorie vergeben wurde). Die *TagVisualization* erstellt das *Tangible*-Objekt, welches folgende graphischen Elemente bei Bedarf erstellt und anzeigt (siehe Abbildung 49):

- *TangibleShape* zum Verbinden mit anderen Tangibles
- *TangibleMenu* zum Auswählen einer Kategorie
- *TangibleSubMenu* zum Einstellen von Parametern bezüglich einer Kategorie
- *DirectConnector* zum Anzeigen von aktiven Verbindungen mit anderen Tangibles
- *LoadingIndicator* zum Anzeigen, dass eine Visualisierung generiert wird

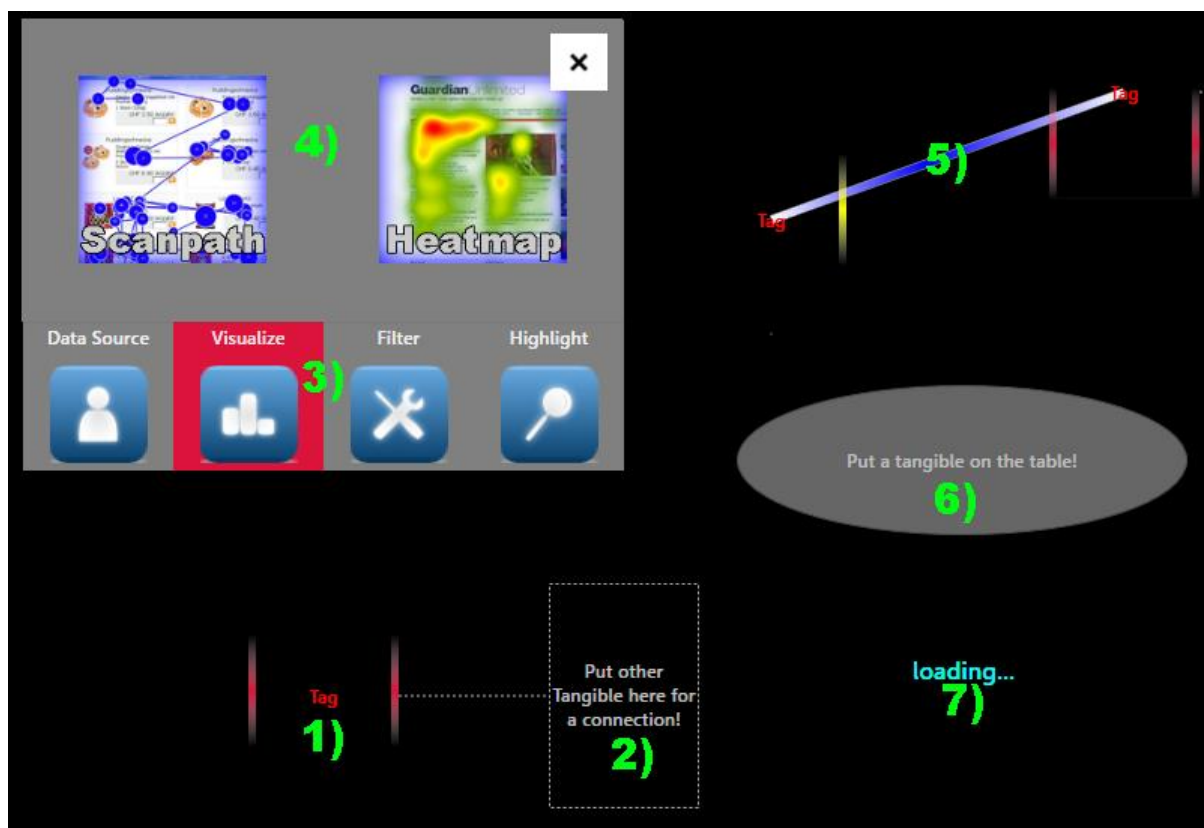


Abbildung 49 – Übersicht über alle graphischen Elemente des Prototyps. 1) Animiertes farbiges Rechteck zur Symbolisierung eines Tangibles auf der Oberfläche. 2) TangibleShape zum Verbinden mit anderen Tangibles. 3) TangibleMenu zum Wählen einer Kategorie. 4) TangibleSubMenu zum Einstellen von Parametern einer Kategorie. 5) DirectConnector zum Anzeigen von aktiven Verbindungen zwischen Tangibles. 6) StartShape als Hilfsmeldung, wenn kein Tangible auf dem Tabletop platziert ist. 7) LoadingIndicator zum Anzeigen, dass gerade eine Visualisierung generiert wird.

Das *TangibleMenu* erlaubt das Setzen einer Kategorie und öffnet das entsprechende *TangibleSubMenu*. Beide blenden sich nach einigen Sekunden Inaktivität automatisch aus. Wird ein *Tangible* innerhalb des *TangibleShapes* platziert, werden die Objekte verbunden und ein *TangibleConnection*-Objekt wird beiden *Tangibles* zugewiesen. Als visuelle Rückmeldung für den Nutzer wird innerhalb der *TangibleConnection* ein *DirectConnector*-Objekt erstellt (siehe Abbildung 50). Seine Animation symbolisiert die Datenflussrichtung gemäß der Regel Input -> Modifier -> Output.

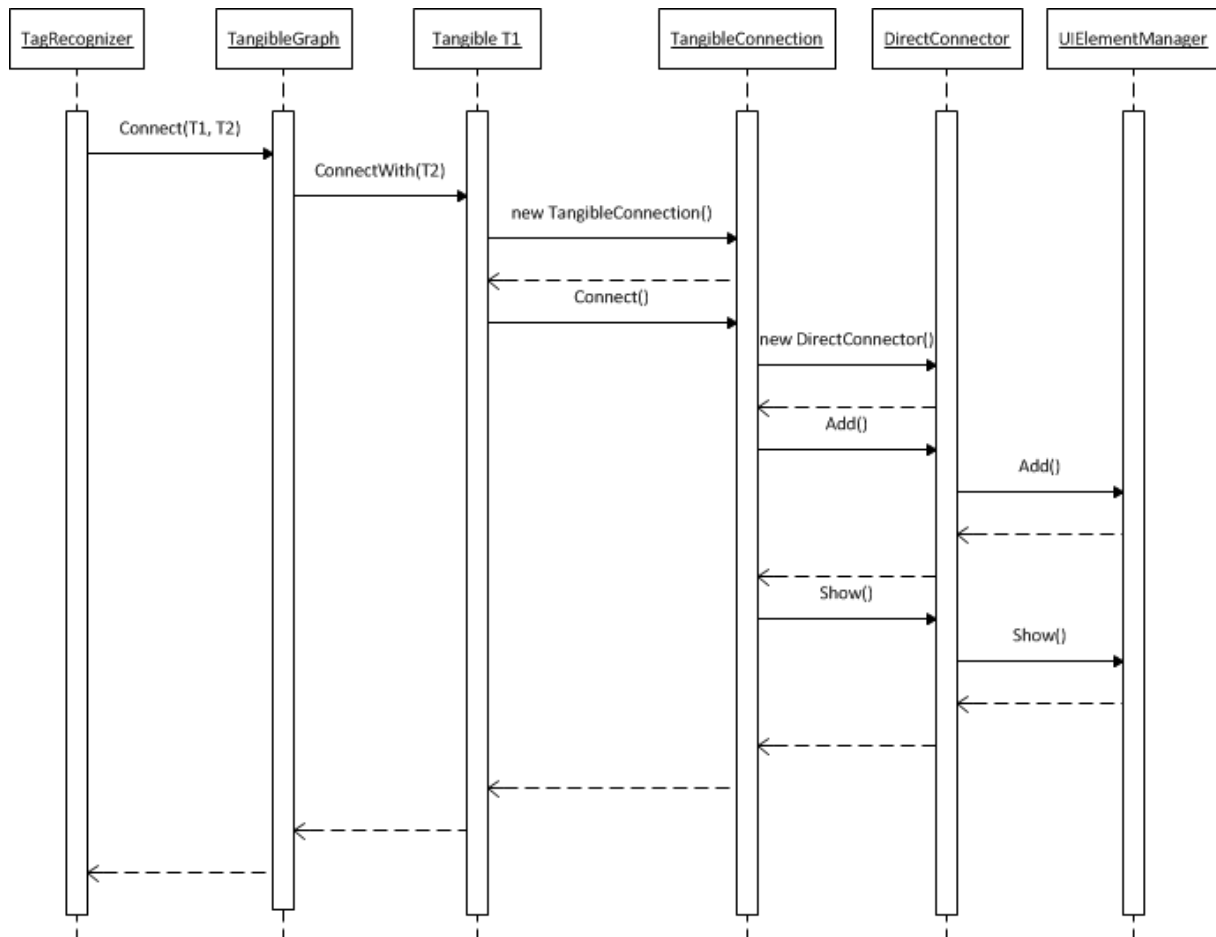


Abbildung 50 – Das Sequenzdiagramm beschreibt den Ablauf beim Verbinden von zwei Tangibles. Interne Fehlerprüfmethode, Zuweisungsmethode und Rückgabewerte sind aus Gründen der Übersichtlichkeit nicht angegeben. Parameter sind nur teilweise angegeben.

Beim Verschieben eines Tangibles werden die Positionen aller *TangibleConnections* permanent aktualisiert und gegebenenfalls neue Verbindungen erstellt, wenn das Tangible in ein kompatibles *TangibleShape* eintritt. Beim Entfernen eines Tangibles werden betroffene Elemente zuerst ausgeblendet und anschließend komplett entfernt.

Das *LoadingIndicator*-Objekt wird erstellt, wenn eine Visualisierung im Hintergrund generiert wird. Es wird oberhalb des Tangibles und unterhalb des eventuell offenen Menüs platziert und verschwindet wieder, sobald die Visualisierung angezeigt wird. An Visualisierungen werden aktuell Heatmaps (*HeatmapVisualization*) und Scan-Path-Diagramme (*ScanPathVisualization*) unterstützt. Über ein Interface (*IVisualization*) können bei Bedarf weitere Visualisierungsformen eingebunden werden. Das Interface definiert folgende Methoden:

- `Display()` – Zeigt eine Visualisierung auf der Tabletop-Oberfläche an
- `Hide()` – Versteckt eine angezeigte Visualisierung
- `Render()` – Generiert eine Visualisierung
- `Remove()` – Entfernt eine Visualisierung
- `Update()` – Aktualisiert eine Visualisierung
- `RenderingCompleted` – Informiert, wenn die Generierung der Visualisierung abgeschlossen ist

7.2.4 Kommunikation

Dieser Abschnitt beschreibt die Kommunikation zwischen Tabletop-Software und den Tangibles auf Arduino-Basis. Als Kommunikationsplattform wird Bluetooth verwendet und zur Implementierung wird eine weit verbreitete Bibliothek [35] benutzt.

Die Klasse *BluetoothServices* speichert eine Zuordnung von Tag Values der Microsoft Byte Tags und Bluetooth-Adressen der Tangibles. Somit weiß die Software, wohin sie ihre Nachrichten schicken muss, um mit dem korrekten Tangible zu kommunizieren. Sollen weitere Tangibles hinzugefügt oder ein Tag geändert werden, muss dies in der Zuordnung dieser Klasse eingefügt werden (siehe Abbildung 51).

```
// Contains a mapping between a Surface Tag ID and the corresponding bluetooth device address
private static readonly Dictionary<int, BluetoothAddress> tagIdTable = new Dictionary<int, BluetoothAddress>()
{
    { 0xC7, BluetoothAddress.Parse("0006664FE52E")},
    { 0xC6, BluetoothAddress.Parse("0006664FDC8B")},
    { 0xC5, BluetoothAddress.Parse("0006664F8ED1")},
    { 0xC4, BluetoothAddress.Parse("0006664FE4F6")}
};
```

Abbildung 51 – Codeauszug des Dictionaries, welches die Zuordnung des Microsoft Byte Tag zur entsprechenden Bluetooth-Adresse des Tangibles speichert. Werden dem System weitere Tangibles hinzugefügt, muss eine neue Zuordnung an dieser Stelle eingefügt werden, damit der Tabletop-Computer mit dem Tangible Nachrichten austauschen kann.

Im Hintergrund läuft permanent ein Mechanismus (siehe Abbildung 52), der auf eingehende Bluetooth-Verbindungen wartet und in der Klasse *BluetoothDeviceManager* implementiert ist. Diese Klasse registriert und verwaltet alle Bluetooth-Geräte der verbundenen Tangibles. Wird ein Tangible auf die Tabletop-Oberfläche gestellt und hat es sich mit der Tabletop-Software verbunden, erstellt der *BluetoothDeviceManager* ein *BluetoothCommunicator*-Objekt und weist es dem *Tangible*-Objekt zu. Das Versenden von Nachrichten verläuft nun über das *BluetoothCommunicator*-Objekt und Tabletop und Tangible können Daten austauschen. Das verwendete Nachrichtenformat ist in Abschnitt 7.1.4 beschrieben.

Nachrichten werden immer dann versendet, wenn das Tangible aktualisierte Informationen erhalten soll, die auf dem Bildschirm angezeigt werden oder wenn der Nutzer den Druckknopf des Tangibles oder die Schüttelgeste benutzt hat.

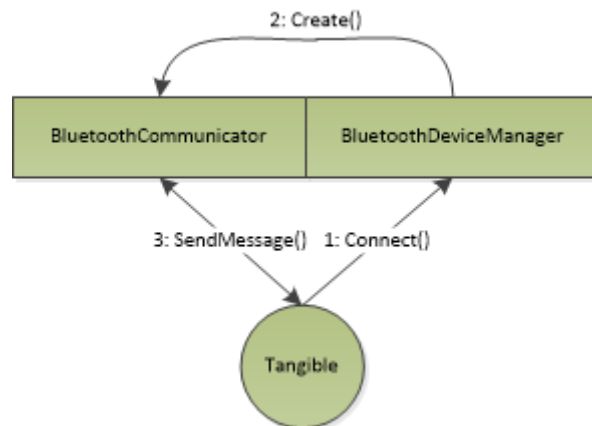


Abbildung 52 – Ablauf bei der Bluetooth-Kommunikation. Zuerst verbindet sich ein Tangible mit dem Listener des BluetoothDeviceManagers und dieser erstellt das BluetoothCommunicator-Objekt. Der Nachrichtenaustausch findet dann zwischen dem BluetoothCommunicator-Objekt und dem Bluetooth-Modul des Tangibles statt.

7.2.5 Graphstruktur und Datenweiterleitung

In diesem Abschnitt wird die Generierung der Graphstruktur aus verbundenen Tangibles sowie die Datenweiterleitung zwischen ihnen erläutert. Die betreffende Klasse heißt *TangibleGraph*.

Ein Graph besteht aus einer Menge von Knoten, die mittels Kanten verbunden sind. Der hier vorgestellte Graph ist gerichtet, es gibt also eine Vorgänger-Nachfolger-Beziehung. Knoten ohne eingehende Kante heißen Wurzeln und Knoten ohne ausgehende Kante heißen Blätter. Je nach Anordnung der Tangibles auf der Tabletop-Oberfläche können mehrere unabhängige Graphen entstehen, die durch Verbinden in einem Graph zusammengefasst werden. Ebenso ist es möglich, durch Trennen von Verbindungen Graphen zu teilen. Abbildung 53 zeigt einen beispielhaften Graphen, wie er durch Tangible-Verbindungen erzeugt werden könnte.

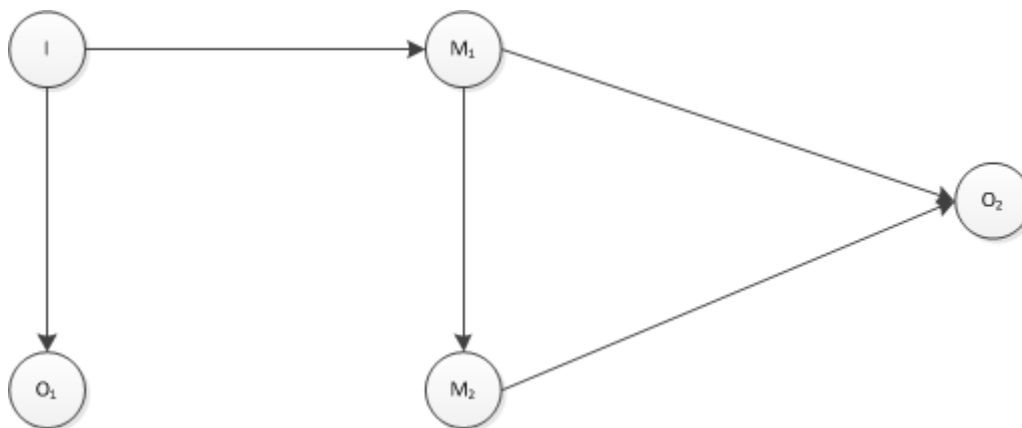


Abbildung 53 – Beispiel einer möglichen Graphstruktur aus verbundenen Tangibles. Der gerichtete Graph besteht aus 5 Knoten (ein Input-Tangible I, zwei Modifier-Tangibles M₁ und M₂ sowie zwei Output-Tangibles O₁ und O₂) und besitzt insgesamt drei Pfade von den Wurzeln zu den Blättern: I -> O₁, I -> M₁ -> O₂ und I -> M₁ -> M₂ -> O₂. Auf diesen Pfaden basiert die Datenweiterleitung. Ein Entfernen von M₁ lässt zwei unabhängige Graphen entstehen.

Die Generierung des Graphen richtet sich nach den Verbindungsregeln (Input -> Modifier -> Output) für Tangibles. So kann jedes Input-Tangible beliebig viele Nachfolger der Kategorien Modifier oder Output haben, aber keinen Vorgänger. Ein Modifier-Tangible kann beliebig viele Vorgänger der Kategorien Modifier und Input und beliebig viele Nachfolger der Kategorien Modifier und Output haben. Ein Output-Tangible kann beliebig viele Vorgänger der Kategorien Modifier und Input haben, aber keinen Nachfolger. In

einem Graph mit mindestens zwei verbundenen Knoten¹³ sind Input-Tangibles immer Wurzeln, Output-Tangibles dagegen immer Blätter. Modifier-Tangibles können sowohl Wurzel als auch Blatt sein, jedoch bringt das keinen Nutzen, da Modifier-Tangibles entweder Daten eines Input-Tangibles benötigen oder ein Output-Tangible, um Daten zur Ausgabe weiterleiten zu können. Sie werden in der Regel immer mindestens einen Vorgänger und einen Nachfolger haben.

Jedes nicht mit anderen Tangibles verbundene Tangible auf dem Tisch wird in einen separaten Graph eingefügt. Werden Tangibles verbunden, werden auch die Graphen intern verbunden. Eine Trennung von Verbindungen führt dazu, dass ein Graph aufgeteilt wird. In der folgenden Abbildung 54 und Tabelle 8 wird das Generieren eines Graphen beschrieben.

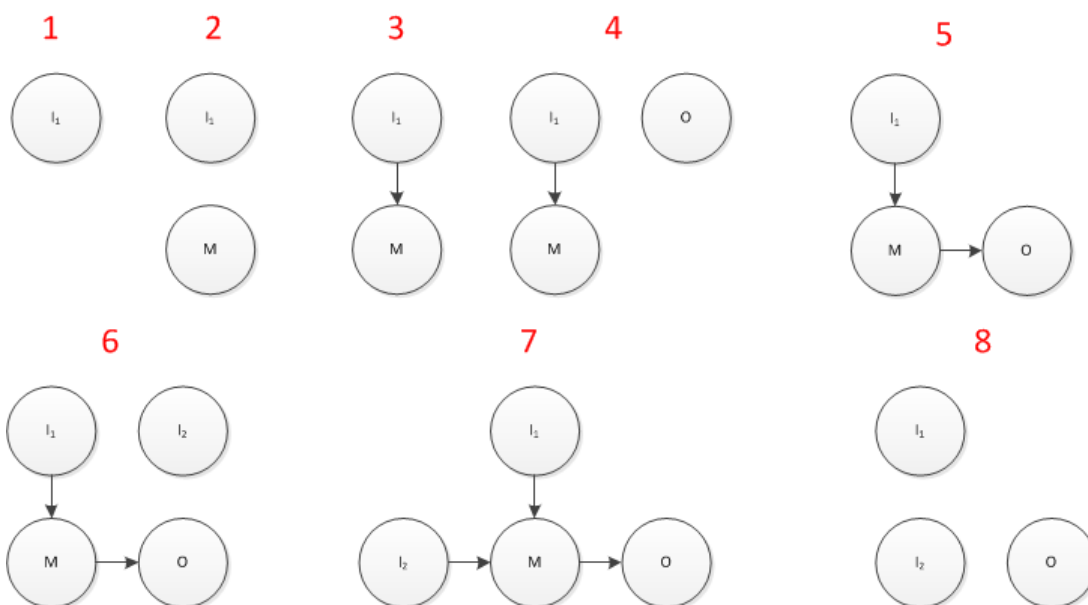


Abbildung 54 – Generierungsbeispiel eines Graphen. Mögliche Operationen sind Einfügen (Schritte 1, 2, 4 und 6), Verbinden (Schritte 3, 5 und 7) und Entfernen (Schritt 8) von Knoten (Tangibles). Am Ende sind drei unabhängige Graphen vorhanden.

Schritt	Aktion	# Knoten	# Kanten	# Graphen
1	Input I_1 eingefügt	1	0	1
2	Modifier M eingefügt	2	0	2
3	I_1 und M verbunden	2	1	1
4	Output O eingefügt	3	1	2
5	M und O verbunden	3	2	1
6	Input I_2 eingefügt	4	2	2
7	I_2 und M verbunden	4	3	1
8	M entfernt	3	0	3

Tabelle 8 – Beschreibung der durchgeführten Schritte in Abbildung 54 zur Erstellung des finalen Graphen. Zu jedem Schritt werden die Anzahl der Knoten, Kanten und Graphen, die die TangibleGraph-Klasse speichert, angegeben.

¹³ Anmerkung: In einem Graph, in dem nur ein Knoten vorhanden ist, ist dieser Knoten sowohl Wurzel als auch Blatt. Daher können Output-Tangibles auch Wurzeln und Input-Tangibles auch Blätter sein.

Die entsprechenden Methoden der Klasse *TangibleGraph* für die Operationen Einfügen, Verbinden und Entfernen sind `Insert()`, `Connect()` und `Remove()`. Die Klasse stellt sicher, dass Knoten nicht doppelt eingefügt werden und aktualisiert beim Verbinden von zwei Knoten automatisch die Benutzeroberfläche.

Um Daten auszugeben, können sie in einer Visualisierung dargestellt werden. Damit die Software weiß, welche Daten innerhalb der Visualisierung dargestellt sind, können mit der *TangibleGraph*-Klasse alle Pfade von den Wurzeln bis zum Blatt (Output-Tangible mit Visualisierung) ermittelt werden (`GetAllPaths()`). Am Beispiel von Abbildung 43 soll nun der Algorithmus, der bestimmt, welche Datensätze aus den Wurzeln (Input-Tangibles) bei einem bestimmten Blatt (Output-Tangible) ankommen, verdeutlicht werden. Im Input-Tangible I links oben sind fünf Datensätze (1-5) aktiviert und im Output-Tangible O₂ rechts soll eine Heatmap generiert werden. Die beiden Modifier-Tangibles M₁ und M₂ sind als Filter konfiguriert, wobei M₁ alles außer Datensatz 2 und 3 filtert und M₂ nur Datensatz 2 durchlässt. Es gibt zwei Pfade vom Input zu diesem Output (I -> M₁ -> O₂ und I -> M₁ -> M₂ -> O₂). Die folgenden beiden Tabellen beschreiben die Mengen für jeden Pfad, die nach dem Passieren eines Tangibles noch vorhanden sind.

Tangible	Einstellungen	Eingangsmenge	Ausgangsmenge
Input I	1-5 aktiv	Alle	{1, 2, 3, 4, 5}
Modifier M₁	2,3 aktiv	{1, 2, 3, 4, 5}	{2, 3}
Output O₂	Heatmap	{2, 3}	{∅}

Tabelle 9 – Datenweiterleitung für Pfad 1 basierend auf dem Graph in Abbildung 53. Aus der Startmenge, die alle verfügbaren Datensätze umfasst, kommen lediglich die Datensätze 2 und 3 am Output an.

Tangible	Einstellungen	Eingangsmenge	Ausgangsmenge
Input I	1-5 aktiv	Alle	{1, 2, 3, 4, 5}
Modifier M₁	2,3 aktiv	{1, 2, 3, 4, 5}	{2, 3}
Modifier M₂	2 aktiv	{2, 3}	{2}
Output O₂	Heatmap	{2}	{∅}

Tabelle 10 – Datenweiterleitung für Pfad 2 basierend auf dem Graph in Abbildung 53. Aus der Startmenge, die alle verfügbaren Datensätze umfasst, kommt lediglich Datensatz 2 am Output an.

Aus der Vereinigung der beiden Mengen, die durch die zwei Pfade geleitet werden, ergibt sich, dass die Heatmap die Datensätze 2 und 3 anzeigen muss.

7.2.6 Datenspeicherung

Da die Tangibles über sehr begrenzte Speicherkapazität sowie Rechenkraft und Batterie verfügen, werden sämtliche Daten über den Status der Tangibles mit der Tabletop-Software verwaltet. Die Software weiß zu jedem Zeitpunkt, welche Daten mit einem Tangible verbunden sind, welche Kategorie es besitzt, mit welchen anderen Tangibles es verbunden ist und welche Visualisierung es anzeigt.

Beim Start der Anwendung sind alle Tangibles im Ausgangszustand. Sie haben keine Kategorie oder assoziierten Daten. Zur Verwaltung der Tangibles wird die Klasse *TangibleManager* benutzt. Sie besitzt eine Liste aller registrierten Tangibles seit dem Start der Anwendung. Beim erstmaligen Setzen des Tangibles registriert die Klasse das Tangible anhand des Tags auf der Unterseite mit der `RegisterTangible()`-Methode. Alle Folgeänderungen am Tangible werden vom Manager registriert und festgehalten. Wird das Tangible vom Tisch entfernt, kennt die Software dennoch ihre Einstellungen und sobald das Tangible wieder auf der Tabletop-Oberfläche platziert wird, weiß die Manager-Klasse über die Methode `IsTangibleKnown()`, dass das Tangible bereits mit Informationen belegt ist und lädt diese.

Auf diese Weise werden die knappen Ressourcen der Tangibles nicht belastet. Die Datenspeicherung ist jedoch nicht permanent, sondern mit dem Beenden des Programms verloren.

8 Prototyp Evaluation

Dieses Kapitel beschreibt die Studie zur Evaluation des Prototyps. Der Prototyp arbeitet mit Eye-Tracking-Daten und soll sich daher in einem Vergleich mit der Analyse-Software von Tobii [1] messen (im Folgenden auch *klassischer Ansatz* genannt). Das Ziel der Evaluation besteht nicht darin, die beiden Software-Systeme zu vergleichen, sondern es soll gezeigt werden, dass mit dem in dieser Arbeit entwickelten Interaktionskonzept ein alternativer Ansatz zur Bewältigung typischer Aufgaben bei der Eye-Tracking-Datenanalyse entstanden ist.

Abschnitt 8.1 beschreibt die Durchführung der Studie, die Aufgabenstellung für die Probanden sowie die Rahmenbedingungen. Die folgenden vier Abschnitte beschreiben den Studienablauf anhand der Aufgabenstellungen. Abschnitt 8.2 fasst die Studienergebnisse zusammen. In Abschnitt 8.3 werden die Schlussfolgerungen aus den Ergebnissen dargelegt und in Abschnitt 8.4 werden Änderungen am Prototyp beschrieben.

8.1 Durchführung

Die Studie wird mit dem in Kapitel 7 vorgestellten Prototyp durchgeführt. Die Software läuft auf einem PixelSense Tabletop-Computer von Microsoft, an dem ein externer Monitor angeschlossen ist. Den Probanden stehen sechs Tangibles zur Verfügung, mit denen das System bedient werden kann. Es stehen zwei funktionsfähige Active Tangibles (siehe Kapitel 7.1) zur Verfügung. Diese werden ergänzt durch drei passive Tangibles. Die passiven Tangibles bestehen aus Plexiglas, sind 5 cm hoch und zylinderförmig. An der Unterseite aller Tangibles ist ein Microsoft Byte Tag angebracht. Mit ihnen kann das System ebenfalls bedient werden, jedoch sind damit nicht alle Möglichkeiten des Prototyps nutzbar (fehlender Bildschirm, kein Druckknopf zum Öffnen des Menüs, kein Beschleunigungssensor, kein Bluetooth). Ein weiteres passives Tangibles ist das Information-Tangible, welches den externen Monitor ansteuert. Dies wird den Probanden zu Beginn der Studie erläutert. Der Aufbau der Studie ist in Abbildung 55 dargestellt.

Die Probanden arbeiten mit Eye-Tracking-Datensätzen der Studie „Visual Elements 2“. Ein Datensatz repräsentiert einen Probanden, der eine Aufgabe ausgeführt hat. Im Prototyp sind insgesamt 23 Datensätze verfügbar.



Abbildung 55 – Aufbau der Evaluationsstudie. Den Probanden stehen sechs Tangibles zur Verfügung. Zwei active Tangibles und drei passive Tangibles (als Gruppe am Rand des Tabletops platziert) und ein Information-Tangible (einzelner Plexiglaszylinder im Vordergrund). Ein externer Monitor ist an den Tabletop-Computer angeschlossen und kann über das Information-Tangible angesteuert werden.

Die Studienteilnehmer sind Experten in der Analyse von Eye-Tracking-Daten. Das bedeutet, dass sie bereits mehrfach mit der zuvor erwähnten Desktop-Software von Tobii Analysen durchgeführt haben. Während der Studie werden die Probanden vier Aufgaben mit dem PixelSense Prototyp lösen und dann anhand von Fragebögen das Konzept bewerten sowie mit der Tobii-Software vergleichen. Die Aufgaben bauen aufeinander auf und sind so gewählt, dass sie sich mit beiden Systemen erledigen lassen.

Zu Beginn erhalten die Teilnehmer eine kurze Erklärung zum Prototyp, zu den erwähnten Einschränkungen und zum Ziel der Studie. Nach der Beantwortung eventuell auftretender Fragen bearbeiten sie selbstständig ohne Hilfe die Aufgaben. Hilfestellung wird nur im Fall gewährt, dass ein Proband eine Aufgabe nicht alleine lösen kann. Zu jeder Aufgabe erhält der Proband zwei Bewertungsbögen: Der erste bewertet die durchgeführte Aufgabe des PixelSense Konzepts, der zweite bewertet das Konzept der Tobii-Software. Beide Fragebögen besitzen eine ähnliche Struktur wie der NASA Task Load Index [36]. Zum Abschluss vergleicht ein letzter Fragebogen die beiden Konzepte direkt miteinander. Die zugehörigen Evaluationsbögen sind im Anhang C und D zu finden. Die durchzuführenden Aufgaben werden nun der Reihe nach vorgestellt.

1) Erzeugen Sie mit dem Prototyp eine Heatmap-Visualisierung, die die 23 Datensätze aller Probanden grafisch ausgibt.

Der Proband muss zwei Tangibles auf dem Tisch platzieren. Im virtuellen Menü des ersten Tangibles wählt er die Kategorie *User Data* und selektiert alle Einträge. Im Menü des zweiten Tangibles wählt er *Visualize* und anschließend *Heatmap*. Sobald die Tangibles verbunden werden, erscheint die Heatmap. Eine exemplarische Lösung ist in Abbildung 56 zu sehen.

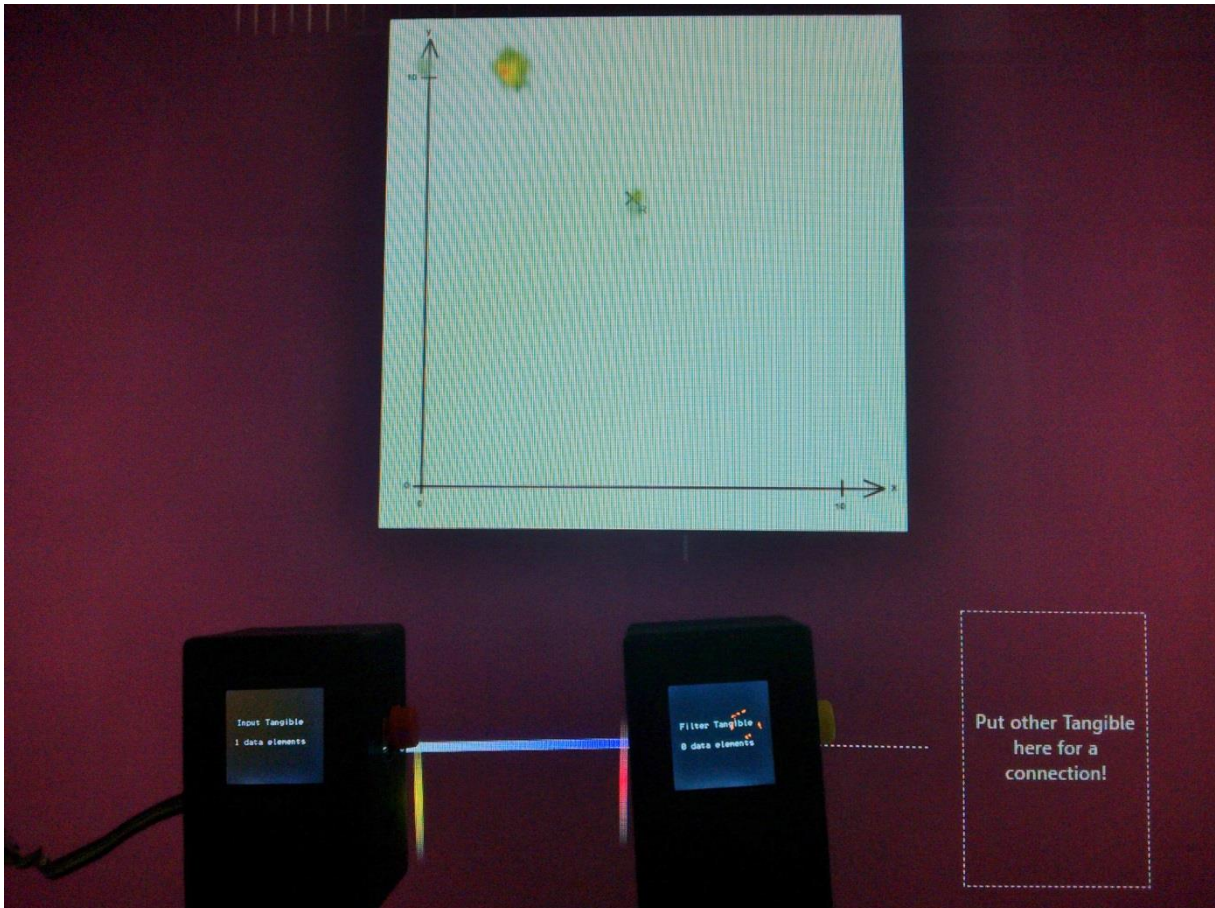


Abbildung 56 – Exemplarische Lösung der ersten Aufgabe der Evaluationsstudie. Ein Input-Tangible (links) und ein Output-Tangible (rechts) sind verbunden, um die Datensätze in einer Heatmap-Visualisierung (oben) anzuzeigen.

2) Entfernen Sie zuerst die Heatmap-Visualisierung. Lassen Sie sich dann anschließend Metainformationen der ersten fünf Probandendatensätze auf dem Informationsbildschirm ausgeben.

Der Proband entfernt das Heatmap-Tangible. Im anderen Tangible ändert er die Auswahl, sodass nur noch die ersten fünf Datensätze selektiert sind. Anschließend verbindet er dieses Tangible mit dem Information-Tangible. Die Metadaten werden auf dem externen Monitor angezeigt. Eine exemplarische Lösung ist in Abbildung 57 zu sehen.

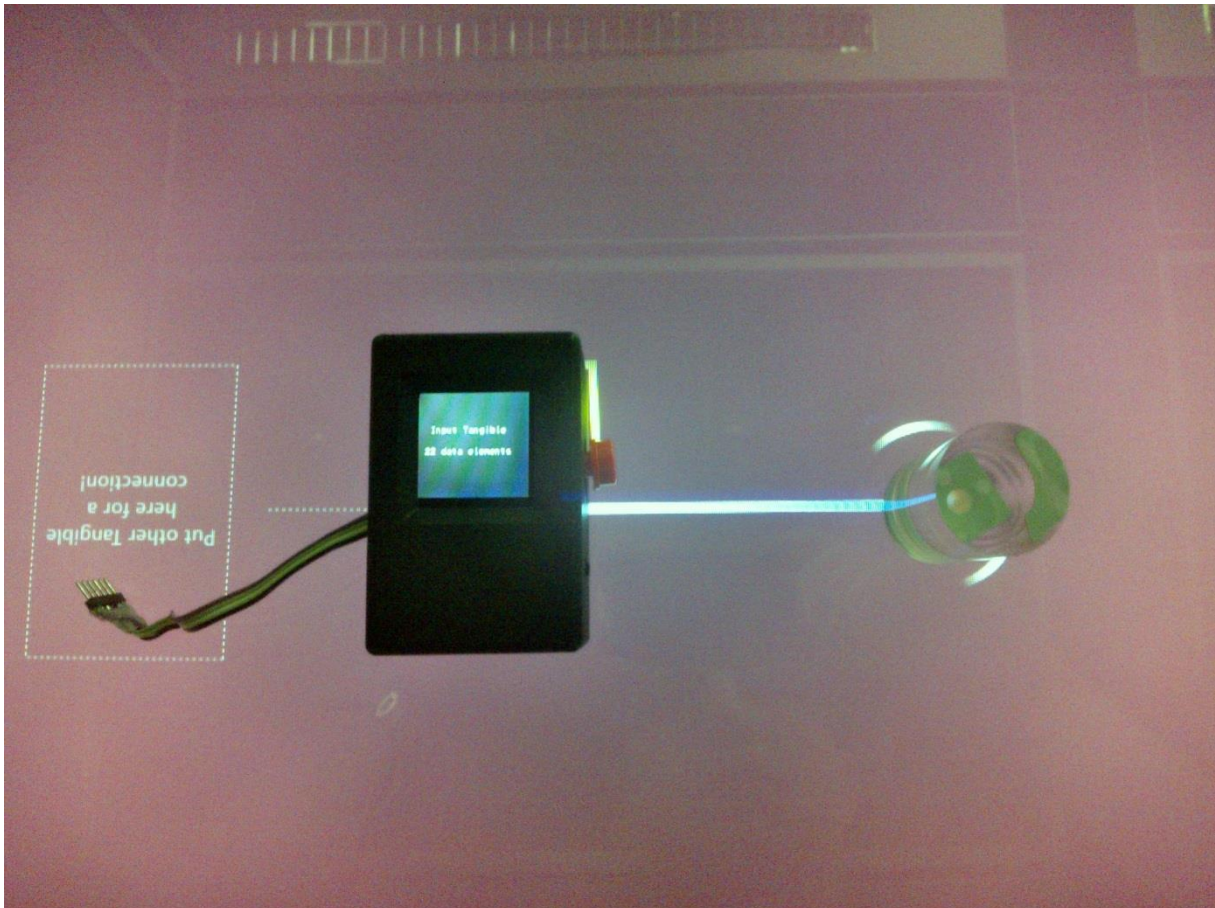


Abbildung 57 – Exemplarische Lösung der zweiten Aufgabe der Evaluationsstudie. Das Input-Tangible mit 23 Probandendatensätzen (links) ist mit dem Information-Tangible (rechts) verbunden. Die Metainformationen der Probandendaten werden auf einem externen Monitor angezeigt.

3) Erzeugen Sie ein Scan-Path-Diagramm, welches die ersten fünf Probandendatensätze darstellt. Die ersten beiden dieser fünf Datensätze sollen hervorgehoben werden.

Der Proband entfernt das Information-Tangible und platziert stattdessen zwei weitere Tangibles auf dem Tisch. Im ersten neuen Tangible wählt er *Highlight* und selektiert die ersten beiden Datensätze. Im zweiten neuen Tangible wählt er *Visualize* und das *Scan-Path-Diagramm*. Zuletzt verbindet er die drei Tangibles, sodass das korrekte Ergebnis angezeigt wird. Eine exemplarische Lösung ist in Abbildung 58 zu sehen.

Diese Aufgabe kann nicht im Vergleich mit dem Tobii-System bewertet werden, da es in der Software keine Möglichkeit gibt, Daten in einem Scan-Path-Diagramm hervorzuheben. Die Bewertung erfolgt daher nur nach den Daten, die der Proband für den Prototyp angegeben hat.

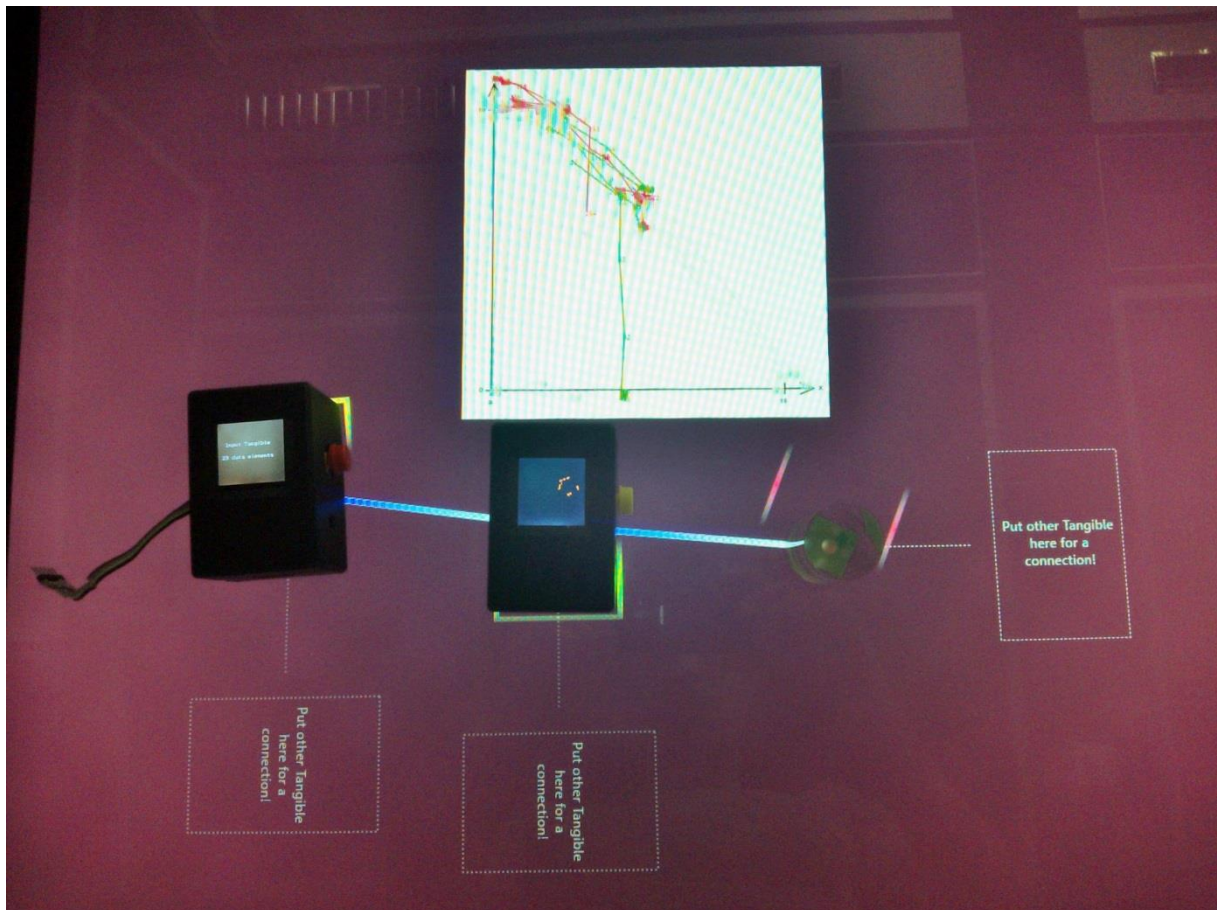


Abbildung 58 – Exemplarische Lösung der dritten Aufgabe der Evaluationsstudie. Ein Input-Tangible (links), ein Modifier-Tangible (mittig) und ein Output-Tangible (rechts) sind miteinander verbunden und generieren die angezeigte Heatmap. Das Input-Tangible leitet fünf Datensätze weiter, das Modifier-Tangible hebt zwei dieser fünf hervor. Das Output-Tangible besteht aus einem Plexiglas-Zylinder.

4) Ändern Sie die Darstellung des Scan-Path-Diagramms, sodass nur noch die beiden Datensätze angezeigt werden, die zuvor hervorgehoben waren. Bringen Sie die Visualisierung anschließend auf den Informationsbildschirm.

Der Proband entfernt das Highlight-Tangible, verbindet die übrigen beiden Tangibles und ändert die Auswahl im Input-Tangible, sodass nur noch zwei Datensätze im Scan-Path-Diagramm angezeigt werden. Zuletzt verbindet er das Information-Tangible mit dem Scan-Path-Tangible. Die Visualisierung wird auf dem externen Monitor angezeigt. Eine exemplarische Lösung ist in Abbildung 59 zu sehen.

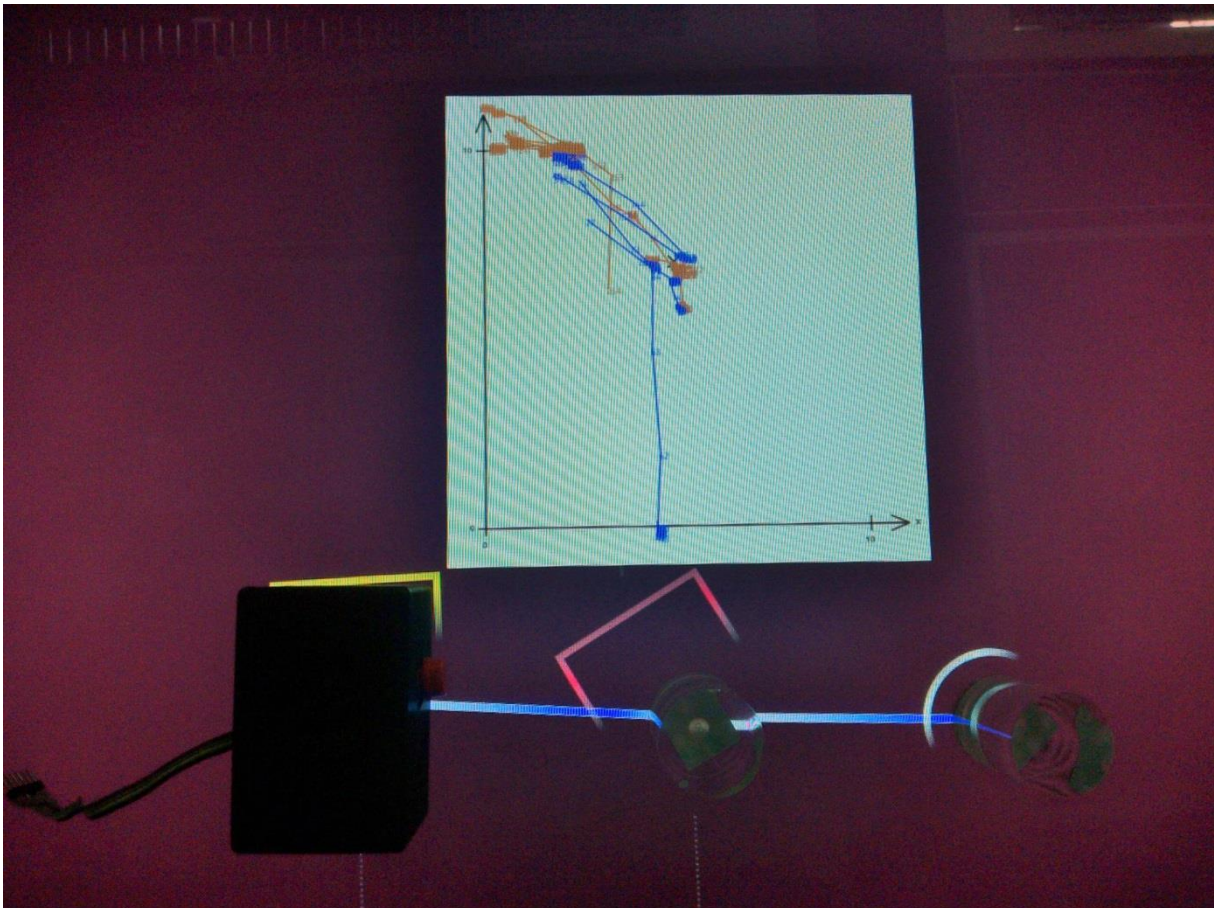


Abbildung 59 – Exemplarische Lösung der vierten Aufgabe der Evaluationsstudie. Ein Input-Tangible (links, gelber animierter Rahmen), ein Output-Tangible (mittig, roter animierter Rahmen) und ein Information-Tangible (rechts, weißer animierter Rahmen) sind verbunden. Die Scan-Path-Visualisierung wird zusätzlich auf einem externen Monitor angezeigt (nicht im Bild).

In den folgenden Abschnitten werden die einzelnen Durchläufe der Studie mit den Probanden beschrieben.

Proband 1

Proband 1 ist männlich, 24 Jahre alt und studiert Informatik. Er kam bei seiner Tätigkeit als studentische Hilfskraft mit der Eye-Tracking-Software von Tobii in Kontakt und hat umfassende Kenntnisse darüber. Er hat bei den ersten beiden Aufgaben Probleme, da er mit dem Interaktionskonzept nicht zurechtkommt. Er benutzt nur ein Tangible und erkennt trotz der Hilfen auf der Tabletop-Oberfläche zunächst nicht, dass die Tangibles in Kombination genutzt werden müssen. Nachdem er das Bedienkonzept durchschaut hat, hat er keine Probleme mit den Lösungen der Aufgaben. Während seiner Studie schalten sich beide Tangibles ab, sodass dem Probanden nur noch die passiven Fähigkeiten (setzen, stellen, verschieben, wegnehmen, drehen) zur Verfügung stehen.

Er bewertet trotz seiner Schwierigkeiten den Prototyp bei den ersten beiden Aufgaben insgesamt besser als den klassischen Ansatz, findet die Bedienung des Prototyps allerdings auch wesentlich anspruchsvoller. Bei längerer Nutzung des Interaktionskonzepts verringert sich auch der Frustrationsgrad des Benutzers, da er besser und erfolgreicher mit dem System arbeiten kann. Im direkten Vergleich ist bei sieben von acht Punkten eine Tendenz Richtung Prototyp zu sehen. Lediglich bei der Frage nach der Flexibilität des Konzepts wählt der Proband eindeutig das Tobii-System.

Diesem Probanden gefällt laut der Datenauswertung das Prototyp-System wesentlich besser als das Tobii-System.

Probandin 2

Probandin 2 ist weiblich, 24 Jahre alt und studiert Mathematik. Sie kam wie Proband 1 während ihrer Tätigkeit als studentische Hilfskraft mit dem Tobii-System in Berührung. Die ersten drei Aufgaben bereiten ihr große Probleme. Sie versucht, alles mit einem Tangible zu bearbeiten anstatt mehrere zu benutzen. Die Hilfsmeldungen und –texte liest sie erst als sie nicht mehr weiter weiß. Danach kann sie die Aufgaben lösen.

Es macht ihr Spaß, mit dem Prototyp zu arbeiten und für Team-Analysen hält sie ihn besser geeignet als das Tobii-System. In allen anderen Kategorien (Frustrationslevel, geistige Anforderung, physische Anforderung,...) und Fragen zum direkten Vergleich der beiden Konzepte präferiert sie jedoch eindeutig den klassischen Ansatz. Bei der Bewertung von Aufgabe 4 ist zwar zu erkennen, dass sie sich mit dem Interaktionskonzept langsam positiver auseinandersetzen kann, allerdings sind die Werte immer noch schlechter als die des Tobii-Systems.

Probandin 3

Probandin 3 ist weiblich, 26 Jahre alt und wissenschaftliche Mitarbeiterin am Institut für Visualisierung und Interaktive Systeme der Universität Stuttgart. Eye-Tracking ist einer ihrer Forschungsbereiche. Aus diesem Grund ist sie sehr vertraut mit dem Software-System von Tobii.

Sie hat ebenfalls mit den ersten Aufgaben einige Probleme, löst die letzten beiden dafür umso leichter, nachdem sie das Interaktionskonzept durchschaut hat. Auch sie benutzt zuerst nur ein Tangible und will ihm mehrere Funktionen nacheinander zuweisen. Das System gefällt ihr und sie probiert auch nach Beendigung der Studie noch einige Dinge damit aus. Sie bemängelt aber gleichzeitig die mangelnde Stabilität des Prototyps, da viele Aufgaben mehrfach ausgeführt werden müssen, bis der Tabletop-Computer eine visuelle Rückmeldung gibt.

Die Auswertung ihrer Angaben weist eine starke Tendenz zum klassischen Ansatz auf. Der Abstand zum Prototyp verringert sich mit jeder Aufgabe, allerdings bleibt der Vorsprung des Tobii-Systems auch dort noch deutlich. Größte Schwächen des Prototyps sind der höhere Frustrationsgrad und die höhere Anstrengung zum Erreichen eines Ziels. Hier bevorzugt die Probandin den klassischen Ansatz. Beim direkten Vergleich der Konzepte punktet der Prototyp bei der Übersichtlichkeit und damit, dass er einfacher zu erlernen ist. Allerdings findet die Probandin den klassischen Ansatz in anderen Punkten besser:

- Mit dem klassischen Ansatz ist es einfacher zu arbeiten
- Der klassische Ansatz gefällt ihr spontan besser
- Der klassische Ansatz ist leichter zu bedienen

Insgesamt kann aus der Datenauswertung eine Tendenz in Richtung des klassischen Ansatzes herausgelesen werden.

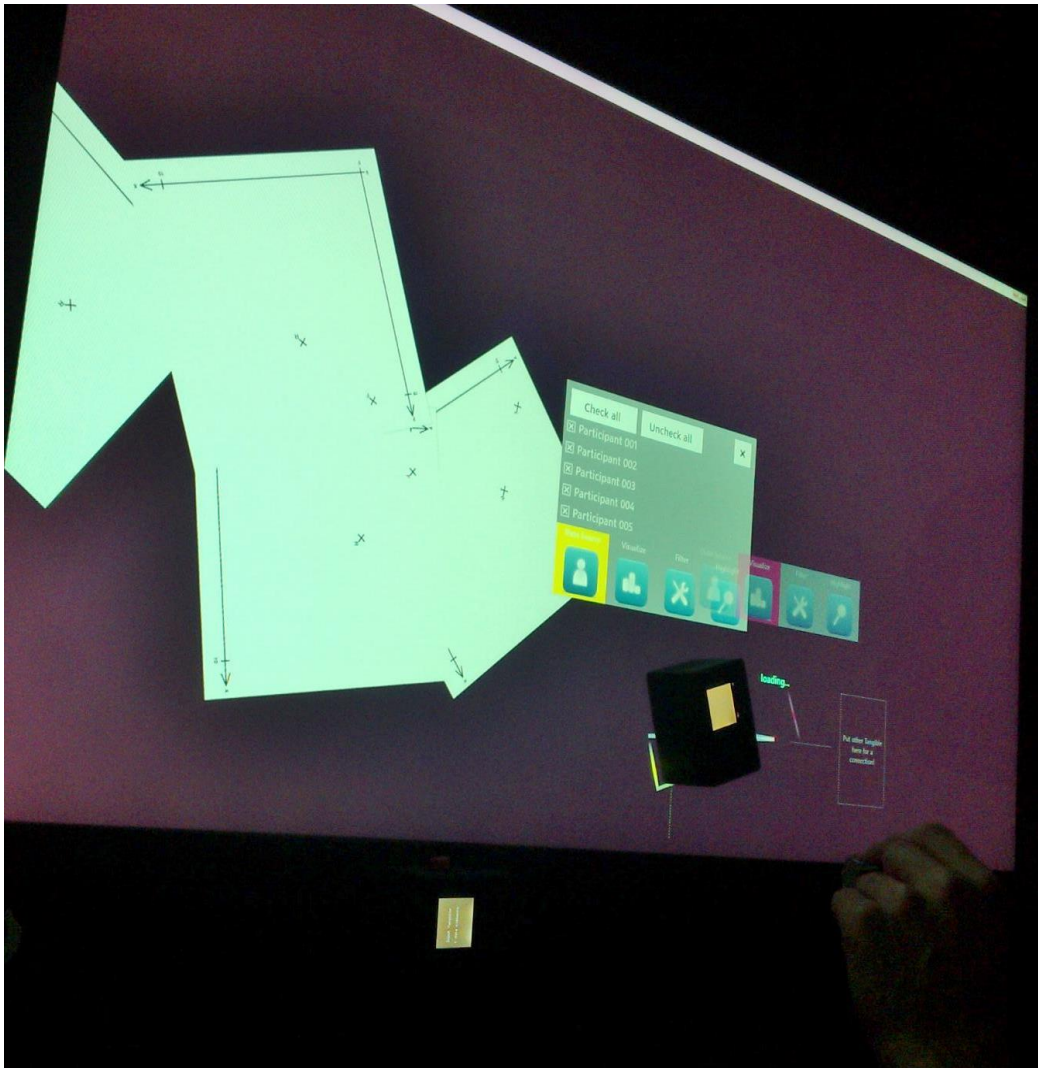


Abbildung 60 – Proband beim Bearbeiten einer Aufgabe während der Evaluationsstudie. Er versucht mit den Tangibles und Touch-Interaktion die ihm gestellte Aufgabe zu lösen.

Proband 4

Proband 4 ist männlich, 30 Jahre alt und wissenschaftlicher Mitarbeiter am Institut für Visualisierung und Interaktive Systeme der Universität Stuttgart. Eye-Tracking gehört zu seinem Forschungsgebiet und deshalb ist er damit sehr vertraut.

Wie alle bisherigen Probanden hat auch er bei der ersten Aufgabe Schwierigkeiten, das Interaktionskonzept zu verstehen. Ab der zweiten Aufgabe hat er keine Probleme mehr und ist sehr schnell mit der Studie fertig. Da der Tabletop-Computer und die Tangibles zum Testzeitpunkt nicht perfekt funktionieren, muss der Proband manche Schritte (etwa das Verbinden von Tangibles) mehrfach ausführen, bis das System reagiert.

Die Auswertung seiner Daten zeigt, dass er bei Aufgabe 1 den Tobii-Ansatz deutlich besser bewertet als den Prototyp. Bei der Bewertung von Aufgabe 4 liegen dann beide in etwa gleich auf. Im direkten Vergleich bevorzugt er je nach Frage mal den Prototyp (bei der Frage nach Team-Analysen und danach, welches Konzept mehr Spaß machen würde) und mal den klassischen Ansatz (der klassische Ansatz ist einfacher und übersichtlicher).

Insgesamt bewertet der Proband die beiden Systeme recht ähnlich, nachdem er einige Zeit mit dem Prototyp gearbeitet hat.

8.2 Zusammenfassung der Studienergebnisse

Eine komplette Übersicht über die Daten, die die Probanden in den Bewertungsbögen des NASA Task Load Index angegeben haben, ist in Tabelle 11 zu finden.

Proband	Aufgabe	Geist. Anforderung	Phys. Anforderung	Zeit. Anforderung	Eigene Leistung	Anstrengung	Frustration	Σ
1	1	14 17	2 1	1 1	2 3	9 18	6 13	34 53
	2	3 14	2 1	7 6	1 1	1 13	8 7	22 42
	3	3 -	3 -	1 -	1 -	4 -	2 -	14 -
	4	14 7	4 1	4 3	1 1	6 12	2 3	31 27
	Σ	34 38	11 3	13 10	5 5	20 43	18 23	101 122
2	1	17 3	5 2	4 2	2 2	12 2	14 2	54 13
	2	11 4	2 2	2 2	2 2	5 3	4 1	26 14
	3	12 -	6 -	9 -	3 -	6 -	12 -	48 -
	4	9 2	3 2	12 2	2 2	9 2	6 3	41 13
	Σ	59 9	16 6	27 6	9 6	32 7	36 6	169 40
3	1	14 8	2 1	1 1	18 4	12 8	12 4	59 26
	2	8 8	2 1	1 1	7 4	6 5	10 3	34 22
	3	5 -	2 -	1 -	3 -	7 -	3 -	21 -
	4	4 4	2 1	1 1	3 3	6 4	8 3	24 16
	Σ	31 20	8 3	4 3	31 11	31 17	33 10	138 64
4	1	18 3	1 1	4 2	9 5	13 2	17 1	62 14
	2	7 6	1 1	2 2	1 1	4 2	2 1	17 13
	3	9 -	1 -	3 -	8 -	5 -	2 -	28 -
	4	2 2	1 1	1 1	1 1	1 1	1 1	7 7
	Σ	36 11	4 3	10 5	19 7	23 5	22 3	114 34

Tabelle 11 – Datenübersicht aller ausgefüllten Bewertungsbögen auf Basis des NASA Task Load Index aller Probanden. Die Skala reicht von 1 bis 20 Punkten pro Antwort, wobei niedrige Werte besser sind. Der erste Wert steht für die Bewertung des Prototyps, der zweite Wert steht für die Bewertung des klassischen Ansatzes. Die hervorgehobenen Zellen summieren die einzelnen Zellen einer Zeile oder Spalte auf. Aufgabe 3 konnte wie in Abschnitt 8.1 beschrieben mit dem Tobii-System nicht durchgeführt werden.

Proband 1 bewertet den Prototyp von allen Teilnehmern am besten (101 Punkte). Aber gleichzeitig bewertet er auch den klassischen Ansatz am schlechtesten (122 Punkte). Bei den anderen Probanden ist der Punkteabstand zwischen Prototyp und Tobii-System deutlich größer und zudem fällt die Bewertung für den Tobii-Ansatz aus. Hier ist ein klarer Vorteil für den klassischen Ansatz zu erkennen. Der Prototyp hat in allen Kategorien über alle Aufgaben hinweg höchstens gleich gute, aber meistens schlechtere Bewertungen als der klassische Ansatz, wenn man von Proband 1 absieht. In der Kategorie Geistige Anforderung ist der Abstand zum klassischen Ansatz über alle Probanden summiert am größten (160 Punkte zu 78 Punkte).

Bei allen Probanden lässt sich ein Lerneffekt beobachten, nachdem sie einige Zeit mit dem Prototyp gearbeitet haben. Dies zeigt sich auch in der Bewertung, da der Prototyp tendenziell in Aufgabe 4 etwas besser bewertet wird als in der ersten Aufgabe.

Frage	\bar{x}	σ
Welches Konzept gefällt Ihnen besser?	+0,25	3,21
Mit welchem Konzept ist es einfacher zu arbeiten?	-1,5	3,61
Welches Konzept ist eher für Analysen im Team geeignet?	+3	3,61
Welches Konzept ist einfacher zu erlernen?	+1	2,58
Welches Konzept ist einfacher zu bedienen?	-0,25	2,08
Welches Konzept ist übersichtlicher?	+1	4,16
Welches Konzept ist flexibler einsetzbar?	-0,75	2,5
Welches Konzept macht mehr Spaß?	+2,5	1,3

Tabelle 12 – Auswertung der Fragebögen zum direkten Vergleich der Interaktionskonzepte des Prototyps mit der Analyse-Software von Tobii. Der Wert in der zweiten Spalte gibt den Durchschnittswert aller Angaben an. Ein Wert zwischen 0 und 4 favorisiert den Prototyp, während ein Wert zwischen 0 und -4 zur Tobii-Software tendiert. Der Wert in der dritten Spalte ist die Standardabweichung. Eine niedrige Standardabweichung bedeutet, dass die Antworten ähnlich sind.

Beim direkten Vergleich der beiden Konzepte (siehe Tabelle 12) zeigt sich, dass die Probanden lieber mit dem Prototyp arbeiten, er ihnen mehr Spaß macht, einfacher zu erlernen und deutlich besser für Teamarbeit geeignet ist als der klassische Ansatz. In den Punkten Bedienung und Flexibilität ist jedoch der klassische Ansatz besser.

8.3 Schlussfolgerungen

Diese Studie zeigt ebenso wie die Vorstudie (siehe Kapitel 6.2), dass den Benutzern ein Einstieg in das Bedienkonzept schwer fällt. Nach einer kurzen Einarbeitungszeit sind sie jedoch mit dem Interaktionskonzept vertraut und können Aufgaben schnell lösen. Dies deutet darauf hin, dass Hilfsfunktionen für das Interaktionskonzept eine wichtige Rolle spielen.

Insgesamt bevorzugen die Teilnehmer das klassische System mit der Desktop-Analyse-Software, auch wenn der Prototyp mehr Spaß macht und für Team-Analysen besser geeignet ist. Die teilweise schlechte Erkennung von Tangibles und Fehlfunktionen des Tabletop-Computers beeinflussen die Bewertung zumindest unterbewusst. Der Prototyp sollte in einem stabilen Zustand einem erneuten Vergleich unterzogen werden, um eine aussagekräftigere Meinung zu erhalten. Zudem sind die Tangibles nicht ausgereift. Die Aktualisierung des Tangible-Bildschirms benötigt mehrere Sekunden, sodass die Probanden sich nicht wirklich sicher sein können, ob die dargestellten Informationen korrekt sind.

Ändert ein Benutzer über das virtuelle Menü die Funktion eines Tangibles, erscheint eine Fehlermeldung. Diese hat sich in der Studie als unbrauchbar erwiesen, da sie von einigen Probanden zwar gelesen wurde, sie ihnen aber keine Hilfe war. Daher sollte diese Meldung geändert oder ersetzt werden.

Die Studie zeigt aber auch, dass das entwickelte Interaktionskonzept trotz Schwächen zur Analyse von Eye-Tracking-Daten eingesetzt werden kann. Alle Probanden konnten alle Aufgaben selbstständig lösen und damit eine einfache Eye-Tracking-Datenanalyse durchführen.

8.4 Nachträgliche Änderungen am Prototyp

Während der Evaluationsstudie konnten einige Probleme beobachtet werden, deren Behebung eine Verbesserung des Interaktionskonzepts erreichen könnte. Die Probleme und Lösungen werden kurz erläutert. Dabei geht es um Hilfsmeldungen, Verbindungen zwischen Tangibles, Farben und um die Aktualisierung des Tangible-Bildschirms.

Hilfsmeldung beim Ändern der Funktion

Die Fehlermeldung beim Ändern der Funktion eines Tangibles über das virtuelle Menü hat sich als unbrauchbar erwiesen. Probanden, die diese Meldung erhielten, empfanden sie nicht als hilfreich. Im Prototyp wird deshalb die Bedienung des virtuellen Menüs geändert.

Wählt der Nutzer eine Funktion, erscheint weiterhin das entsprechende virtuelle Untermenü. Allerdings werden nun alle anderen Funktionen ausgegraut und deaktiviert. Der Nutzer kann sie wieder aktivieren, indem er erneut auf den Menüeintrag klickt, der die aktuelle Funktion repräsentiert, um diesen abzuwählen. Nun sind alle Funktionen (Menüelemente) wieder auswählbar. Der Nutzer muss sich in diesem Fall bewusst dafür entscheiden, eine Funktion zu deaktivieren, um eine andere aktivieren zu können. Dadurch kann das Bewusstsein dafür, dass Tangibles nur eine Funktion repräsentieren sollen, erhöht werden.

Tangible-Verbindungen

Während der Evaluationsstudie stellten Nutzer oftmals ein Tangible ohne Funktion in die Umriss eines bereits auf dem Tisch platzierten Tangibles, um eine Verbindung zu initiieren (siehe Abbildung 37, Kapitel 6.3.6). Sie wählten anschließend eine Funktion und dachten, dass die Tangibles verbunden seien. Allerdings können nur kompatible Tangibles verbunden werden und solange ein Tangible keine Funktion besitzt, ist es mit allen anderen inkompatibel. Deswegen wird ein Mechanismus eingebaut, der bei der Auswahl einer Funktion über das virtuelle Menü prüft, ob das Tangible innerhalb der Umriss eines anderen Tangibles steht. Ist dies der Fall und sind die beteiligten Tangibles kompatibel, wird eine Verbindung hergestellt.

Farben

Die Animationsfarben der Tangibles sind für Benutzer des Systems zwar sichtbar, aber sie erkennen keinen Zusammenhang zwischen einer Farbe und einer Tangible-Kategorie. Deshalb wird eine Farblegende auf dem externen Bildschirm angezeigt. Sollte das Information-Tangible benutzt werden, wird die Farblegende ausgeblendet. Zusätzlich wird die Farbe als Hintergrund des Tangible-Bildschirms angezeigt, sodass Nutzer mit einem kurzen Blick die Funktion des Tangibles sehen können.

Aktualisierung des Tangible-Bildschirms

Die Aktualisierung des Tangible-Bildschirms dauert zu lange und wird manchmal nicht korrekt ausgeführt, da Nachrichten verloren gehen oder falsch interpretiert werden können. Aus diesem Grund wird das Nachrichtenformat um eine fortlaufende Nummer ergänzt. Jede Nachricht wird von der Gegenseite anhand dieser Nummer identifiziert und gegebenenfalls eine Bestätigung oder Fehlermeldung zurückgeschickt.

9 Zusammenfassung

Das Ziel der Arbeit war es, ein Datenanalysekonzept auf Basis von Tabletop-Computern und Tangible User Interfaces zu entwickeln, das von mehreren Personen gleichzeitig benutzt werden kann. Zusätzlich zum Konzept sollte am Beispiel von Eye-Tracking-Daten ein Prototyp-System mit eigens gebauten Tangibles entwickelt werden.

Zu Beginn wurde eine Sammlung von Anforderungen durchgeführt, um die gewünschten Eigenschaften des Interaktionskonzepts zu ermitteln. Dies geschah durch Beobachtungen von Eye-Tracking-Analysevorgängen sowie durch Gespräche mit Experten.

Auf Basis der Anforderungen wurde ein initiales Interaktionskonzept entwickelt und mittels Paper Prototyping in einer Benutzerstudie evaluiert. Das Ziel der Studie bestand darin, das Konzept zu verbessern sowie Fehler und Anmerkungen zu erhalten. Die Studienergebnisse flossen in die finale Version des Konzepts ein. Kapitel 6 beschreibt die Vorüberlegungen, die Studie und das finale Konzept.

Das Prototyp-System, das auf dem entwickelten Interaktionskonzept basiert, wurde anschließend implementiert. Dabei wurde einerseits eine Anwendung für den Microsoft PixelSense Tabletop-Computer auf Basis von C#, dem .NET-Framework 4.5 sowie dem Surface SDK 2 entwickelt. Des Weiteren wurden Tangibles zuerst mit dem .NET Micro Framework und dem .NET Gadgeteer Kit und später dann mit Hilfe des Arduino-Systems konzipiert, gebaut und programmiert. Der Prototyp arbeitet beispielhaft mit Eye-Tracking-Daten, ist jedoch generisch gehalten und kann ebenso mit einer anderen Datenbasis benutzt werden.

Zuletzt wurde eine zweite Studie durchgeführt, um den Prototyp (und damit das entwickelte Interaktionskonzept) zu verifizieren. Da der Prototyp mit Eye-Tracking-Daten arbeitet, wurden Eye-Tracking-Experten hinzugezogen, um einen Vergleich zwischen dem Prototyp und dem klassischen Analysekonzept für Eye-Tracking-Daten anhand der Analyse-Software von Tobii machen zu können.

9.1 Bewertung

Dieser Abschnitt befasst sich mit einzelnen Aspekten dieser Arbeit und bewertet die Vorgehensweisen, aufgetretenen Probleme und gefällten Entscheidungen.

Tabletop-Computer

Der verwendete Tabletop-Computer PixelSense von Microsoft ist aufgrund seiner Bauweise und dem mitgelieferten Surface SDK vielseitig einsetzbar, jedoch hat er auch einige Schwachstellen. Er ist anfällig für Infrarotlicht in der Umgebung (spezielle Lampen oder Sonnenstrahlen) und produziert bei längerem Betrieb viel Hitze. Diese beiden Effekte sorgen dafür, dass die Erkennung von Objekten auf der Oberfläche nicht mehr zuverlässig funktioniert. Objekte, die sich auf dem Tisch befinden, werden entweder regelmäßig als neue Objekte oder überhaupt nicht erkannt. Dies erschwert die Nutzung eines Tangible User Interfaces teilweise erheblich. Ist die Umgebung um den Tisch relativ dunkel, funktioniert die Erkennung auch merklich besser.

Interaktionskonzept

Das finale Interaktionskonzept soll für mehrere Arten von Datenanalysen verwendet werden können und nicht nur auf Eye-Tracking-Daten wie im Prototyp beschränkt sein. Dennoch ist die Fokussierung auf Eye-Tracking während der Entwicklung sehr stark gewesen. Es wurden Eye-Tracking-Datenanalysen beobachtet, es wurde mit Eye-Tracking-Experten diskutiert und

es wurden Eye-Tracking-Daten als Basis für die Studien genutzt. Um die Tauglichkeit des Analysekonzepts für andere Datenmodelle zu verifizieren, sollten Studien durchgeführt werden.

Implementierung

Die Trennung zwischen Hard- und Software (Tangibles und Tabletop) während der Implementierung führte nicht automatisch zu zwei individuellen Entwicklungsphasen.

Für die Tabletop-Software gibt es keinen umfangreichen Software-Entwurf. Vielmehr ist die Software schrittweise entstanden, in dem neue Funktionen eingebaut wurden, sobald die bisherigen erfolgreich getestet wurden. Deswegen ist die Erweiterbarkeit des Systems nicht optimal. Um dies zu erreichen, hätte von Anfang an ein Mechanismus (etwa ein Plug-In-System) integriert werden müssen.

Tangibles

Die selbstgebauten Tangibles benötigen zu viel Zeit, um ihre Informationen auf dem Bildschirm zu aktualisieren. Es können mehrere Sekunden vergehen, bis die Ansicht aktualisiert ist. In der Zwischenzeit weiß der Benutzer nicht, ob die Anzeige korrekt ist und er kann sich daher nicht auf die Daten verlassen. Da es keine softwareseitige Fehlerüberprüfung gibt, kann es auch passieren, dass Nachrichten verlorengehen und der Bildschirm eines Tangibles überhaupt nicht aktualisiert wird.

Insgesamt sind die Tangibles zu fehleranfällig. Die modulare Bauweise, bei der alle Teile ohne Werkzeug austauschbar sind, macht die Tangibles wesentlich weniger robust und die Funktion des gesamten Tangibles kann bereits durch das minimale Verrutschen einer Komponente beeinträchtigt werden.

Vorstudie

Die Vorstudie mittels Paper-Prototyping lieferte zwar viele wichtige Erkenntnisse zum finalen Entwurf eines Interaktionskonzepts, allerdings haben 75% der Teilnehmer einen starken Bezug zur Informationstechnik durch ihr Studium oder ihren Beruf. Daneben sind sie alle zwischen 24 und 30 Jahre alt. Für eine repräsentativere Studie sollten die Probanden deshalb vielseitiger in Bezug auf Alter, Geschlecht und Bildungsstand sein, um bessere Ergebnisse zu erhalten. Dennoch haben die Probanden zahlreiche nützliche Hinweise und Verbesserungsvorschläge für das Interaktionskonzept geliefert.

Evaluationsstudie

Aufgrund der sehr speziellen Anforderungen an die Probanden konnte die Studie nur mit einem kleinen Teilnehmerkreis durchgeführt werden. Außerdem wurde eine Aufgabe gestellt, die mit der Tobii-Software erfolgreich bearbeitet werden kann. Die technischen Probleme mit dem eingesetzten Tabletop-Computer und den Tangibles haben die Bewertung durch die Probanden ebenfalls beeinflusst. Die Ergebnisse sind hilfreich, aber dennoch sollte eine angemessenere Studie (mehr Teilnehmer, Aufgaben müssen mit beiden Konzepten gelöst werden) das Interaktionskonzept dieser Arbeit verifizieren.

10 Ausblick

Dieses Kapitel beschreibt mögliche Erweiterungen des entwickelten Interaktionskonzepts, wie es verbessert werden könnte und wie es in bestehende Arbeiten integriert werden könnte.

Benutzerstudien

Das in dieser Arbeit entwickelte Interaktionskonzept ist mit Eye-Tracking-Daten getestet worden. Da es sich aber um ein allgemeines Interaktionskonzept für Daten aller Art handeln soll, muss dies durch weitere Studien überprüft werden. Im Rahmen dieser Arbeit konnten lediglich eine Studie zur Entwicklung des Interaktionskonzepts sowie eine Evaluationsstudie auf Basis von Eye-Tracking-Daten durchgeführt werden.

Vor allem die Evaluationsstudie wurde mit einem sehr kleinen Teilnehmerkreis durchgeführt. Eine weitere Studie in größeren Rahmen würde bessere Vergleichswerte liefern.

Tabletop und Tangibles

Das User Interface mit selbstgebauten Tangibles und dem Tabletop-Computer PixelSense funktioniert, leidet aber unter einigen Schwächen:

- PixelSense leidet unter Infrarot-Strahlung in der Umgebung
- PixelSense produziert und sendet selbst Infrarot-Strahlung bei längerer Laufzeit aus
- Tangible-Gehäuse muss aus nicht reflektierendem Material bestehen, damit die Reflexionen den PixelSense nicht beeinträchtigen
- PixelSense erkennt Objekte (etwa einen Arm oder eine Hand) in geringem Abstand über der Tischoberfläche, wodurch unabsichtlich Aktionen ausgeführt werden können
- PixelSense erkennt Objekte mit geringer aber spürbarer Verzögerung

Um diese Probleme teilweise zu beheben, könnte ein Tabletop-Computer mit anderer Technik, die etwa weniger anfällig für Infrarot-Strahlung in der Umgebung ist (siehe Kapitel 2.3.1), verwendet werden.

Die in Kapitel 2.2 erwähnten Sifteo Cubes können als Tangibles benutzt werden. Sie verfügen über einen Bildschirm, Sensoren, Kommunikationsmöglichkeiten und sind programmierbar, sodass sie theoretisch in das Interaktionskonzept integriert werden können. Da sie kompakter als die für diese Arbeit entwickelten Tangibles sind, stünde mehr Platz auf der Tabletop-Oberfläche zur Verfügung.

Externe Geräte

Eine sinnvolle Erweiterung des Konzepts wäre die Einbindung von zusätzlichen Geräten. Ein rudimentärer Ansatz ist durch das Ansteuern eines externen Monitors bereits aktuell möglich. Durch Integration von Tablets und Smartphones ließe sich die Benutzeroberfläche erweitern und auch teilen. So könnte etwa ein Analyst mit seinem Tablet nahe an den Tabletop-Computer treten, sodass die Benutzeroberfläche erweitert wird. Er verschiebt eine dargestellte Visualisierung auf sein Tablet und verlässt den Tisch wieder. Die Oberfläche wird wieder verkleinert und der Analyst kann sich die Daten der Visualisierung auf seinem Gerät ansehen. Auf diese Weise können sich auch Personen vom Tisch entfernen und trotzdem noch am Analyseprozess teilnehmen.

Einbindung in Powerwall Interactions

Pfleging et al. [24] (siehe Kapitel 3) veröffentlichten ein System zur kollaborativen Datenanalyse mithilfe von Powerwalls und Tangible User Interfaces. Dabei können zusätzliche Geräte wie etwa Tablets integriert werden, um an der Datenanalyse teilzunehmen. Das hier entwickelte Interaktionskonzept könnte prototypisch als eine Art Moderator einer kollaborativen Analysesitzung fungieren. Jedes Gerät über einen „öffentlichen Raum“ verfügt, der über alle Geräte synchronisiert wird. Somit ließen sich die am Tabletop-Computer generierten Informationen einfach auf alle angeschlossenen Geräte verteilen. Eine kleine Analystengruppe bereitet am Tabletop mithilfe des Tangible User Interfaces Visualisierungen von bestimmten Datensätzen vor und diese werden anschließend auf die anderen Geräte synchronisiert. So ließe sich die Arbeit in kleinere Pakete aufteilen.

Evolution der Interaktion

Die nächste Evolutionsstufe könnten *Virtual Tangible User Interfaces* sein. Bei einem Tangible User Interface interagiert man mit Objekten der realen Welt, bei Virtual Tangible User Interfaces interagiert man mit virtuellen Objekten. Ein virtuelles Objekt könnte etwa eine 3D-Projektion im Raum sein, die über Gesten manipulierbar ist (siehe Abbildung 61). Mit dieser Interaktionsform wäre die Benutzeroberfläche nicht mehr auf den Tabletop-Computer beschränkt, sondern könnte den ganzen zur Verfügung stehenden Raum benutzen. Tangibles müssten nicht mehr real sein, sondern man arbeitet mit Projektionen und Gesten. Die Materialkosten werden verringert, es gibt keinen Verschleiß von Teilen mehr und die virtuellen Objekte können genau das darstellen, was gerade benötigt wird. Für das hier entwickelte Interaktionskonzept ergeben sich dadurch neue Möglichkeiten. Die Analyse wird nicht mehr durch den Tisch beschränkt. Die Anzahl der zur Verfügung stehenden Tangibles ist unbegrenzt. Die Steuerung der virtuellen Tangibles kann über Gesten erfolgen, sodass eine neue Interaktionsmöglichkeit in das Konzept integriert werden kann. Eine kollaborative Analyse muss nicht mehr an einem Raum stattfinden. Die Synchronisierung der Tangibles kann über das Internet erfolgreich, womit Teamarbeit und Telearbeit besser verschmelzen können. Die Zukunft bietet spannende Möglichkeiten.



Abbildung 61 – Szene aus dem Film Iron Man. Der Protagonist arbeitet mit einem virtuellen Objekt, welches er über Gesten manipulieren kann. Quelle: themashable.com¹⁴

¹⁴ <http://thesmashable.com/wp-content/uploads/2012/06/Marvel-The-Avengers-Movie-2012-HD-Wallpaper-Iron-Man-Tony-Stark-12.jpg>

Anhang A: Startfragebogen der Vorstudie

Angaben zur Person

Geschlecht: Männlich Weiblich

Alter:

Beruf:

Vorkenntnisse

Wie schätzen Sie Ihre Kenntnisse beim Analysieren von Daten ein, die in Eye-Tracking-Studien aufgezeichnet wurden?

Anfänger Fortgeschritten Experte

Besitzen Sie ein Smartphone mit touch-fähigem Bildschirm (z. B. iPhone)?

Ja Nein

Wenn ja, wie lange nutzen Sie das Gerät täglich in etwa?

Bis 30 Minuten 31 bis 60 Minuten 1-2 Stunden über 2 Stunden

Besitzen Sie einen Computer mit touch-fähigem Bildschirm (z. B. Convertible)?

Ja Nein

Wenn ja, wie lange nutzen Sie das Gerät täglich in etwa?

Bis 30 Minuten 31 bis 60 Minuten 1-2 Stunden über 2 Stunden

Besitzen Sie einen Tablet Computer (z. B. iPad)?

Ja Nein

Wenn ja, wie lange nutzen Sie das Gerät täglich in etwa?

Bis 30 Minuten 31 bis 60 Minuten 1-2 Stunden über 2 Stunden

Haben Sie generell Erfahrung mit der Bedienung von Geräten, die über einen berührungsempfindlichen Bildschirm verfügen?

Ja Nein

Haben Sie mit einem Tabletop Computer gearbeitet (z. B. Microsoft PixelSense)?

Ja Nein

Haben Sie in diesem Zusammenhang auch mit Tangibles gearbeitet?

Ja Nein

Falls ja, bitte beschreiben Sie in kurzen Worten das Szenario, in dessen Zusammenhang Sie mit Tangibles in Berührung kamen.

Anhang B: Bewertungsbogen der Vorstudie

War das Lösen der Aufgaben einfach oder kompliziert?

Einfach

Kompliziert

War es einfach oder schwer, einen Einstieg in die Bedienung des Systems zu finden?

eher einfach

eher schwer

War es einfach oder schwer, das Bedienkonzept des Systems zu verstehen?

eher einfach

eher schwer

Waren die Hilfestellungen des Systems ausreichend, wenn Sie nicht weiterkamen?

Ausreichend

nicht ausreichend

War die Funktionsweise der Tangibles leicht oder schwer ersichtlich?

Leicht ersichtlich

Schwer ersichtlich

War die Anzahl der zu benutzenden Tangibles angemessen?

Zu wenige Tangibles

Angemessen

Zu viele Tangibles

Hat das Arbeiten mit dem System Spaß gemacht?

Hat keinen Spaß gemacht

Hat viel Spaß gemacht

Würden Sie gerne auch andere Systeme mit Tangibles bedienen?

eher nein

eher ja

Anhang C: Bewertungsbogen NASA-TLX der Evaluationsstudie

Aufgabennummer:

Benutztes System:

Geistige Anforderung

Wie hoch schätzen Sie Ihre geistige Anforderung beim Ausführen der Aufgabe ein?



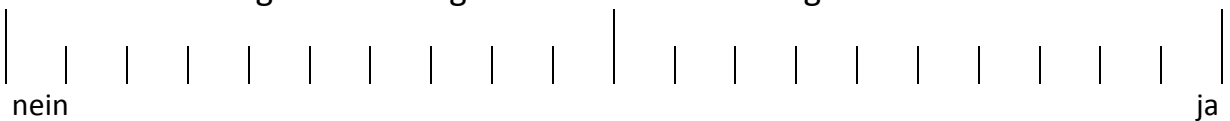
Physische Anforderung

Wie körperlich anstrengend fanden Sie die Aufgabe?



Zeitliche Anforderung

Trieb Sie die Aufgabenstellung bei ihrer Durchführung zur Eile?



Eigene Leistung

Wie erfolgreich waren Sie beim Durchführen der Aufgabe?



Anstrengung

Wie hart mussten Sie arbeiten, um Ihre Ergebnisleistung zu erreichen?



Frustration

Wie verunsichert, entmutigt, irritiert, gestresst oder verärgert waren Sie über die Aufgabe?



Anhang D: Vergleichsfragebogen der Evaluationsstudie

Die folgenden Fragen sollen einen Vergleich zwischen dem hier vorgestellten Interaktionskonzept zur Analyse von Eye-Tracking-Daten und dem klassischen Analysekonzept ermöglichen.

1) Welches Konzept gefällt Ihnen spontan besser?

Prototyp				gleich					klassischer Ansatz

2) Mit welchem Konzept ist es einfacher zu arbeiten?

Prototyp				gleich					klassischer Ansatz

3) Welches Konzept ist eher für Analysen im Team geeignet?

Prototyp				gleich					klassischer Ansatz

4) Welches Konzept ist einfacher zu erlernen?

Prototyp				gleich					klassischer Ansatz

5) Welches Konzept ist einfacher zu bedienen?

Prototyp				gleich					klassischer Ansatz

6) Welches Konzept ist übersichtlicher?

Prototyp				gleich					klassischer Ansatz

7) Welches Konzept ist flexibler einsetzbar?

Prototyp				gleich					klassischer Ansatz

8) Mit welchem Konzept macht es Ihnen mehr Spaß zu arbeiten?

Prototyp				gleich					klassischer Ansatz

Literaturverzeichnis

- [1] tobii, *Eye tracking products*. Available: <http://www.tobii.com/en/eye-tracking-research/global/products/> (2013, Aug. 13).
- [2] Microsoft Corporation, *The Power of PixelSense*. Available: <http://www.microsoft.com/en-us/pixelsense/pixelsense.aspx> (2013, Apr. 29).
- [3] H. Ishii and B. Ullmer, "Tangible bits: towards seamless interfaces between people, bits and atoms," in *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, New York, NY, USA: ACM, 1997, pp. 234–241.
- [4] M. Weiser, "The computer for the 21st century," *SIGMOBILE Mob. Comput. Commun. Rev*, vol. 3, no. 3, pp. 3–11, <http://doi.acm.org/10.1145/329124.329126>, 1999.
- [5] Edwin Püttmann, "Tabletop-Computer-basierte Steuerung für Powerwall-Visualisierungen," Diplomarbeit, Visualization and Interactive Systems Institute (VIS), Stuttgart, Germany, 2012.
- [6] Bill Buxton, *Multi-Touch Systems that I Have Known and Loved*. Available: <http://www.billbuxton.com/multitouchOverview.html> (2013, Apr. 22).
- [7] Wayne Westerman, "HAND TRACKING, FINGER IDENTIFICATION, AND CHORDIC MANIPULATION ON A MULTI-TOUCH SURFACE," University of Delaware, Delaware, 1999.
- [8] R. J. Jacob, A. Girouard, L. M. Hirshfield, M. S. Horn, O. Shaer, E. T. Solovey, and J. Zigelbaum, "Reality-based interaction: a framework for post-WIMP interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA: ACM, 2008, pp. 201–210.
- [9] D. Wigdor and D. Wixon, *Brave NUI world: Designing natural user interfaces for touch and gesture*. Burlington, MA: Morgan Kaufmann, 2010.
- [10] Patrick Frigg, *Tangible User Interfaces: Beispiele und eine Taxonomie*. Zürich.
- [11] D. Merrill, E. Sun, and J. Kalanithi, "Sifteo cubes," in *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts*, New York, NY, USA: ACM, 2012, pp. 1015–1018.
- [12] J. Schöning, P. Brandl, F. Daiber, F. Echtler, O. Hilliges, J. Hook, M. Löchtfeld, N. Motamedi, L. Muller, P. Olivier, T. Roth, and U. von Zadow, "Multi-Touch Surfaces: A Technical Guide," *Technical Report TUM-I0833, Technical Reports of the Technical University of Munich, (2008)*, 2008.
- [13] J. Y. Han, "Low-cost multi-touch sensing through frustrated total internal reflection," in *Proceedings of the 18th annual ACM symposium on User interface software and technology*, New York, NY, USA: ACM, 2005, pp. 115–118.
- [14] Microsoft Corporation, *Tagged Object Integration for Surface 2.0*. Available: <http://www.microsoft.com/en-us/download/details.aspx?id=28959> (2013, Aug. 02).
- [15] *Discover the Samsung SUR40*. Available: <http://www.microsoft.com/en-us/pixelsense/WhatsNew.aspx> (2013, May. 02).
- [16] A. Valli and L. Linari, "Natural interaction sensitivetable," in *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, New York, NY, USA: ACM, 2008, pp. 2315–2318.
- [17] Andreas Gebhardt, "Grundlagen des Rapid Prototyping: Eine Kurzdarstellung der Rapid Prototyping Verfahren," *RTejournal*, vol. 1, pp. 1–16, <http://opus.bibliothek.fh-aachen.de/opus/volltexte/2006/160/>, 2004.

- [18] Isabel Unger, *3D-Drucker: Die Technologie im Überblick*. Available: <http://t3n.de/magazin/3d-druck-schicht-schicht-zukunft-230252/> (2013, May. 02).
- [19] Arduino, *Arduino*. Available: www.arduino.cc (2013, May. 02).
- [20] N. Villar, J. Scott, S. Hodges, K. Hammil, and C. Miller, "NET gadgeteer: a platform for custom devices," in *Proceedings of the 10th international conference on Pervasive Computing*, Berlin, Heidelberg: Springer-Verlag, 2012, pp. 216–233.
- [21] S. Taylor, S. Izadi, D. Kirk, R. Harper, and A. Garcia-Mendoza, "Turning the tables: an interactive surface for vjing," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA: ACM, 2009, pp. 1251–1254.
- [22] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner, "The reactTable: exploring the synergy between live music performance and tabletop tangible interfaces," in *Proceedings of the 1st international conference on Tangible and embedded interaction*, New York, NY, USA: ACM, 2007, pp. 139–146.
- [23] J. Patten, H. Ishii, J. Hines, and G. Pangaro, "Sensetable: a wireless object tracking platform for tangible user interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA: ACM, 2001, pp. 253–260.
- [24] Bastian Pfleging, Michael Raschke, Edwin Püttmann, and Nico Ploner, "Powerwall Interactions," in *CHI 2013 Extended Abstracts*
- [25] H.-C. Jetter, J. Gerken, M. Zöllner, H. Reiterer, and N. Milic-Frayling, "Materializing the query with facet-streams: a hybrid surface for collaborative search on tabletops," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA: ACM, 2011, pp. 3013–3022.
- [26] Edwin Püttmann, "Tabletop-Computer-basierte Steuerung für Powerwall-Visualisierungen," Visualization and Interactive Systems Institute (VIS);, Stuttgart, Germany, 2012.
- [27] Nico Ploner, "Gestensteuerung für Powerwall-basierte Visualisierungen," Visualization and Interactive Systems Institute (VIS), Stuttgart, Germany, 2012.
- [28] J. Ludewig and H. Lichter, *Software Engineering: Grundlagen, Menschen, Prozesse, Techniken*, 2nd ed. Heidelberg: dpunkt-Verl, 2010.
- [29] C. Snyder, *Paper prototyping: Fast and easy way to design and refining user interfaces*. San Francisco, Calif, Oxford: Morgan Kaufmann; Elsevier Science, 2003.
- [30] M. Raschke, X. Chen, and T. Ertl, "Parallel scan-path visualization," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, New York, NY, USA: ACM, 2012, pp. 165–168.
- [31] Florian Haag, Michael Raschke, and Thomas Ertl, "Adaptable Filter Graphs - Towards Highly-Configurable Query Visualizations," in *Informatik 2012*, pp. 1059–1073.
- [32] Sijie Wang, "Comparing Tangible and Multi-touch Interfaces for a Spatial Problem Solving Task," Master Thesis, Simon Fraser University, Burnaby, 2008.
- [33] David Schmid, "Einfache Herstellung durch natürliche Interaktion mit Lasercuttern im Heimbereich," Institut für Visualisierung und Interaktive Systeme, Universität Stuttgart, Stuttgart, Germany, 2013.
- [34] *BluTack*. Available: <http://www.blutack.com> (2013, Nov. 20).
- [35] In The Hand Ltd, *32feet.NET - Personal Area Networking for .NET*. Available: <http://32feet.codeplex.com/> (2013, Nov. 13).
- [36] *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2011.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Stuttgart, den 22.11.2013 _____