

Institut für Visualisierung und Interaktive Systeme  
Universität Stuttgart  
Universitätsstraße 38  
70569 Stuttgart  
Germany

Diplomarbeit Nr. 3440

**Evaluierung des Einflusses  
verschiedener Interaktionsmodalitäten  
auf die Merkfähigkeit**

Erkan Hoser

**Studiengang:** Informatik

**Prüfer:** Prof. Dr. Albrecht Schmidt

**Betreuer:** Dr. Florian Alt, Stefan Schneegaß M.Sc.

**begonnen am:** 22.01.2013

**beendet am:** 24.07.2013

**CR-Klassifikation:** H.5.2

---

---

## Zusammenfassung

In der heutigen Zeit müssen wir uns sehr viel merken. Doch ist es oft schwer die aufgenommenen Informationen zu behalten. In dieser Arbeit wird untersucht, wie Interaktion die Merkfähigkeit beinflusst. Zu diesem Zweck wurde ein Spiel für Android Smartphones entworfen und implementiert, das mithilfe von fünf verschiedenen Interaktionsformen gespielt werden kann. Diese Interaktionsformen sind *Keine Interaktion*, *Geäusche*, *Annäherung*, *Schütteln* und *Telefon drehen*. In einer Laborstudie haben 30 Personen mithilfe dieser Interaktionsformen interagiert. Die Spieldaten wurden dabei zur späteren Auswertung gespeichert. Außerdem wurde für jede Interaktionsform die User Experience gemessen

Die Ergebnisse lassen den Schluss zu, dass die Interaktion mithilfe des Annäherungssensors des Smartphones für die Merkfähigkeit am Besten ist. Die Interaktionsform, bei der man das Smartphone schütteln muss, schneidet am Schlechtesten ab. Bei der User Experience ist es am Besten, wenn man außer dem Berühren des Touchscreens keine besondere Interaktionsform einsetzt. Das Schütteln des Smartphones hat auch bei der User Experience die schlechtesten Werte erzielt.

## Abstract

Nowadays we have to memorize a lot of things. This, however is not always easy. This thesis studies how memorability is influenced by interaction techniques. Therefore, a game for Android smartphones is designed and implemented, with which users can interact using five different interaction techniques. These techniques are *No interaction*, *Volume*, *Proximity*, *Shake* and *Turn phone*. The game was tested in a lab study by 30 participants. The game data was recorded for later evaluation. Additionally the User Experience for each interaction technique was determined.

The results lead to the conclusion, that the interaction with the proximity sensor of the device has the best influence on memorability. Shaking the device as an interaction technique has the worst results. For the best user experience one had to use no particular interaction technique except touching the screen. Shaking the device has the worst result in user experice too.



# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>7</b>
<b>2 Verwandte Arbeiten und Hintergrund</b>	<b>11</b>
2.1 Gesteneingabe . . . . .	11
2.1.1 Touchscreen . . . . .	11
2.1.2 Sensoren . . . . .	15
2.1.2.1 Bewegungs- und Beschleunigungssensor . . . . .	15
2.1.2.2 Positionssensoren . . . . .	18
2.1.3 Kamera . . . . .	19
2.2 Audio . . . . .	21
2.2.1 Sprache . . . . .	21
2.2.2 Geräusche . . . . .	22
2.3 Interaktion und Merkfähigkeit . . . . .	24
<b>3 Konzept</b>	<b>27</b>
<b>4 Implementierung</b>	<b>33</b>
4.1 Tools . . . . .	33
4.2 Architektur . . . . .	33
4.2.1 Client . . . . .	35
4.2.2 Server . . . . .	48
<b>5 Studie</b>	<b>51</b>
<b>6 Ergebnisse der Studie</b>	<b>55</b>
6.1 Einfluss auf die Merkfähigkeit . . . . .	55
6.2 Ergebnisse des User Experience Fragebogens . . . . .	60
<b>7 Zusammenfassung und Ausblick</b>	<b>63</b>
<b>8 Anhang</b>	<b>65</b>

## INHALTSVERZEICHNIS

---

# 1 Einleitung

Merkfähigkeit ist die Fähigkeit eines Menschen sich an zuvor aufgenommene Informationen zu erinnern. Merkfähigkeit ist in der heutigen Gesellschaft sehr wichtig, sei es im Beruf oder in der Freizeit. Man hat Termine, an die man sich erinnern muss, Telefonnummern, die man behalten möchte, Dinge, die man erledigen muss, und vieles mehr. Dabei ist es nicht immer leicht, sich Informationen zu merken. Der Mensch hat nur ein begrenztes Kurzzeitgedächtnis, und nur durch bestimmte Methoden (wie Wiederholung) kann man Informationen längerfristig behalten. Doch was ist mit anderen Methoden, mit denen man sich Dinge besser merken kann? Visuelle und akustische Reize können dabei helfen. Hilfestellung geben auch verschiedene Merkhilfen wie zum Beispiel Eselsbrücken, Assoziationsketten und andere. Gibt es aber vielleicht Methoden, die noch nicht oder nur wenig erforscht sind?

Heutzutage werden wir von allen Seiten mit Informationen bombardiert, sei es erwünschte Information, wie beispielsweise Nachrichten, der Wettervorhersage, oder auch unerwünschte Information, wie Werbung. Diese Informationen werden uns auf verschiedenen Medien wie Zeitung, Bücher und E-Mails präsentiert. Die traditionellen (d.h. gedruckten) Medien werden immer mehr durch elektronische Medien verdrängt. Einen Großteil der Informationen wird auf Bildschirmen, Tablets und Smartphones angezeigt. Erwünschte Informationen wollen wir uns dabei natürlich merken.

Doch dies ist im Alltag nicht immer einfach. Wie oft kommt es vor, dass man sich einmal angeschauten Informationen nicht dauerhaft merken kann? Sei es die Nummer des Abfahrtsgleises eines Zuges, die Uhrzeit zu der die Geburtstagsparty eines Freundes beginnen soll, oder die Hausnummer einer Adresse, zu der man das erste Mal unterwegs ist. Wenn man die Information nur einmal zu sehen bekommt, ist es oft schwierig sich diese längere Zeit zu merken. Man sucht beispielsweise in einem Bürokomplex eine bestimmte Person. Im Erdgeschoss ist ein Register mit den Namen der Mitarbeiter mit den zugehörigen Raumnummer. Vielleicht kann man sich das Stockwerk merken, aber wenn man im richtigen Stockwerk angekommen ist, hat man die Raumnummer schon wieder vergessen. Wäre es einfacher gewesen sich die Information zu merken, hätte man mit der Information interagieren müssen? Wenn man anstatt eines Registers ein elektronisches Display mit Touchscreen nutzen würde, hätte man die Möglichkeit die Raumnummer erst dann anzuzeigen, wenn der Besucher auf den Namen des Mitarbeiters klickt.

Man stelle sich vor, man ist in einer fremden Stadt und will an eine bestimmte Adresse. Nach einem Herumirren findet man ein öffentliches Display mit einer Stadtkarte. Man merkt sich den Weg zum Ziel und läuft los. Doch nach kurzer Zeit schon, ist man sich nicht sicher, ob links oder rechts abgebogen werden musste. Wäre es vielleicht besser gewesen, es hätte die Möglichkeit gehabt den Weg auf der Karte mit den Fingern nachzuzeichnen, so dass dieser farbig unterlegt

wird? Hätte die Interaktion mit dem Display dabei geholfen, sich den Weg besser zu merken?

In den vorangegangenen Beispielen hat man situationsbedingt nicht die Möglichkeit die Information noch einmal anzufordern, es sei denn man will zurück zum festen Informationspunkt. Doch was ist mit mobilen Geräten? Auch hier gibt es Situationen, in denen dies nicht möglich ist.

Man hat eine wichtigen Geschäftssitzung mit ausländischen Partnern. Um die Gegenseite zu beeindrucken hat man mit Hilfe eines Smartphones die Begrüßung samt Aussprache gelernt. Doch was ist, wenn es soweit ist, und man sich im entscheidenden Moment nicht daran erinnern kann? Man kann schlecht das Smartphone auspacken und nachschauen. Man sieht also dass man sich trotz Mobilgeräte Dinge zuverlässig merken muss. In einer interaktiven Form dieses Lernens, hätte man vielleicht das Wort selbst nachsprechen können. Mit Hilfe von Spracherkennungssoftware hätte die korrekte Aussprache verifiziert werden können. Um sich die Schreibweise einzuprägen, hätte man zum Beispiel das Wort mit dem Finger auf dem Bildschirm des Smartphones nachzeichnen können.

An diesen drei Beispielen kann man nun die Fragestellung der Arbeit gut erkennen, welche lautet : *Hilft Interaktion dabei uns Dinge besser zu merken?*

Um diese Fragestellung zu beantworten, wurde als Gerät, mit dem man interagieren muss, das Smartphones ausgesucht. Es wird oft im Alltag genutzt, ist mobil, handlich, und man kann damit auf vieler Weise interagieren. Dies wird durch die große Zahl von Sensoren ermöglicht. Diese wären unter Anderem der Touchscreen, die Kamera oder der Bewegungssensor.

Doch man führt die Interaktion mit dem Gerät natürlich nicht ohne Grund aus, man braucht also eine Motivation. Ein für das Smartphone implementiertes Spiel ist dank des Spieltriebs des Menschen hervorragend geeignet.

Bei dem ausgewählten Spiel sollte es um die Merkfähigkeit gehen. Die Wahl fiel auf ein dem bekannten Memory ähnliches Spiel. Dieses Spiel kann gut mit verschiedenen Interaktionsformen kombiniert werden. Man kann diese dann vergleichen und bei dem Gewinner gegebenenfalls weitere Untersuchungen durchführen. Diese Untersuchungen können die Verfeinerungen der Interaktionsform und den Einflus auf verschiedene Arten von Merkfähigkeit beinhalten.

Das Spiel wurde also mit mehreren Interaktionsformen implementiert. Die Untersuchung fand in Form einer Laborstudie statt, bei der die Teilnehmer in eine kontrollierte Umgebung eingeladen wurden und das Spiel mit den verschiedenen Interaktionsformen spielen mussten. Die Spieldaten wurden dann automatisch an einen Server geschickt, der diese gespeichert hat. Später wurden sie dann ausgewertet.

---

Der Aufbau dieser Arbeit wird im Folgenden beschrieben.

Im 2. Kapitel wird ein Überblick über verwandte Literatur geschaffen. Hier werden vor allen verschiedene Interaktionsformen betrachtet. Zum Schluss des Kapitel werden auch Arbeiten zur Merkfähigkeit aufgezeigt.

Im 3. Kapitel wird das Konzept ausführlicher erklärt. Es wird erklärt wie die Untersuchung durchgeführt wurde und wieso diese Form der Durchführung gewählt wurde.

Im 4. Kapitel wird näher auf den Implementierungsteil eingegangen. Es werden die implementierte Software und die dafür eingesetzten Tools betrachtet.

Das 5. Kapitel beschäftigt sich mit der Studie und deren Durchführung. Es wird genau erklärt wie die Studie durchgeführt wurde.

Im darauf folgenden 6. Kapitel werden die Ergebnisse der Studie betrachtet und diskutiert.

Im 7. Kapitel, wird die Vorgehensweise zusammengefasst und die Ergebnisse werden nochmals in einer knappen Form präsentiert. Zusätzlich gibt es noch eine Diskussion über Verbesserungen und mögliche zukünftige Forschungsthemen.

Das 8. Kapitel ist der Anhang.



---

## 2 Verwandte Arbeiten und Hintergrund

In diesem Abschnitt wird auf verschiedene, mit dem Thema dieser Arbeit verwandten Arbeiten, eingegangen. Die meisten dieser Arbeiten beschäftigen sich dabei mit Gesten, seien es zweidimensionale Gesten, die auf den Touchscreen eines Gerätes ausgeführt werden, oder dreidimensionale Gesten, also solche, die alle drei Raumachsen ausnutzen. Bei dreidimensionalen Gesten kann man entweder das Gerät selbst zur Ausführung der Geste nutzen, oder Gesten mit den Sensoren des Gerätes erkennen lassen. Im ersten Fall werden Bewegungs- und Beschleunigungssensoren verwendet, für den zweiten Fall die Kamera, oder auch das Magnetometer. Bei Gesten, die im Raum um das Gerät ausgeführt werden, wird auch von Around Device Interaction[27] gesprochen.

Eingabeformen, bei denen keine Gesten verwendet werden können, sind solche, die Sprache und Geräusche nutzen. Diese werden unter dem Punkt Audio zusammengefasst.

Zum Schluss des Kapitels wird noch auf die Merkfähigkeit allgemein, und in Verbindung mit Interaktion eingegangen.

### 2.1 Gesteneingabe

Gesten sind Körperbewegungen eines Menschen, die primär zur Verständigung dienen. Kelly und Barr haben durch verschiedene Experimente versucht, die Bedeutung von Gesten zum Verständnis der Absicht einer Person zu verstehen [21]. Dabei mussten Teilnehmer Videos mit zwei Personen, die sich auf verschiedene Weise miteinander verständigen, anschauen. Die Verständigung konnte dabei mit Sprache, einer Kombination von Sprache und Gesten, oder nur Gesten ablaufen. Die Teilnehmer mussten bei jedem Video angeben, ob sie die Intention des letzten Satzes, ein indirekte Anfrage, verstanden haben. Kelly und Barr kamen zu dem Schluss, dass Sprache und Gestik zusammen zum Verständnis der Kommunikation beitragen.

Gesten die bei der Interaktion mit Smartphones von Benutzern favorisiert werden sind beispielsweise solche, die eine Metapher für eine Handlung aus der Realität darstellen [34].

#### 2.1.1 Touchscreen

Die primäre Interaktionsart für mobile Geräte ist die Nutzung des Touchscreens. Praktisch alle Smartphones und Tablets sind mit einem solchen ausgestattet. Die Verwendung von Maus und Tastatur ist bei diesen Geräten entweder nicht möglich, oder zumindest nicht vorgesehen. Aber wie gut kommen die Menschen mit den Touchscreen zurecht? Nicolau und Jorge haben untersucht, wie ältere Menschen mit einem Touchscreen zurechtkommen [33]. Der Touchscreen wurde zur Texteingabe genutzt und es wurde die Geschwindigkeit und Genauigkeit gemessen. Außerdem wurde das Zittern der Hand und dessen Einfluss auf das

Tippverhalten analysiert. Zusätzlich wurden auch die verschiedenen Fehlertypen untersucht.

Der häufigste Fehlertyp war das Weglassen von Buchstaben, gefolgt von falsch eingegebenen und zusätzlich eingegebenen Buchstaben. Das von Nicolau und Jorge gemessene Zittern der Hand korreliert mit den Fehlern.

Findlater et al. haben den Unterschied der Performanz von jungen und alten Leuten bei der Eingabe mit Maus einerseits, und einem Touchscreen andererseits untersucht [13]. Die Ergebnisse zeigten dass, alte Leute im Allgemeinen zwar langsamer waren als junge, die Performanzunterschiede aber durch den Touchscreen verringert wurden. Die jüngeren Teilnehmer konnten durch den Einsatz des Touchscreens eine Steigerung der Performanz um 16% erzielen, bei den Älteren lag diese bei 35%.

Ein Problem von Touchscreens ist die Präzision der Eingabe. Albinsson und Zhai haben zwei Techniken, welche ein schnelles, effizientes und pixelgenaues Zeigen ermöglichen sollen, entwickelt[1]. Ihre Technik Precision-Handle hatte gute Werte bei Geschwindigkeit, Genauigkeit und Komfort. Ihre Technik Cross-Keys ist wahrscheinlich sehr exakt und daher geeignet für sehr feine Änderungen.

Bei der Eingabe mit Touchscreens kann man zwischen singletouch, also Einagabe mit einem, und multitouch, also Eingabe mit mehreren Fingern unterscheiden. Kin et al. haben die Unterschiede zwischen der Eingabe mit einer Maus, singletouch, multitouch mit je einem Finger pro Hand, und multitouch mit mehreren Fingern untersucht[26]. Dabei war die schnellste multitouch Eingabe etwa doppelt so schnell, wie die Eingabe mit der Maus. Singletouch war durchschnittlich für 83% der Verringerung der Auswahlzeit verantwortlich. Die Bedingung, bei der ein Finger pro Hand genutzt wurde, machte den Rest der Geschwindigkeitsverbesserung aus. Kin et al. haben auch herausgefunden, dass es für Benutzer, die keine Erfahrung mit multitouch haben, keine Rolle spielte, ob sie einen Finger pro Hand oder mehrere Finger einsetzten.

Donahue et al. haben die Performanz bei der Problemlösung mit Maus, Touchscreen und einem Tangible User Interface(TUI), also einer anfassbaren Benutzerschnittstelle, untersucht[11]. Diese drei Eingabemöglichkeiten unterscheiden sich in ihrer “Nähe”. Die Ergebnisse zeigten, dass das TUI signifikant langsamer als Maus und Touchscreen ist. Jedoch waren TUI und Touchscreen bei der Problemlösung wesentlich besser. Insgesamt bat der Touchscreen die beste Kombination aus Geschwindigkeit und Effizienz.

Doch der Touchscreen hat auch Nachteile. Im Gegensatz zu anfassbaren Tasten, gibt es meistens kein taktiler, also fühlbares Feedback. Hoggan et al. haben die fingerbasierte Texteingabe bei Geräten mit Touchscreen untersucht[18]. Dabei haben sie ein Gerät mit physikalischen Tasten, eines mit einem Touchscreen ohne taktilem Feedback und ein Gerät mit einem Touchscreen mit taktilem Feedback, verglichen. Die Ergebnisse zeigen, dass taktiler Feedback die Performanz bei der fingerbasierten Texteingabe wesentlich verbessert und nahe an die Performanz von einer physikalischen Tastatur heranbringt.

Ein anderer Nachteil ist natürlich die zweidimensionale Eingabe. Doch auch hier

kann man etwas tun. So haben Brewster und Hughes die druckbasierte Texteingabe auf Touchscreens untersucht[9]. Ein sanftes Tippen auf dem Bildschirm sollte dabei kleingeschriebene und ein festes Tippen großgeschriebene Buchstaben erzeugen. Es wurden zwei verschiedene druckbasierte Tastaturen mit einer Standardtastatur mit Shift Taste untersucht. Die Teilnehmer wurden beim Sitzen und beim Gehen beobachtet. Brewster und Hughes haben festgestellt, dass druckbasierte Eingabe die Eingabe mit einer normalen Tastatur mit Shift Taste bei der mobilen Texteingabe übertreffen kann.

Heo und Lee haben den Druck auf den Touchscreen genutzt um den Modus bei Ziehoperationen zu ändern[17]. In der Desktopumgebung würde man für das Ziehen die Maus, und für das ändern des Modus eine Taste benutzen. Heo und Lee haben ForceDrag entworfen, eine Ziehoperation, bei der Druck zum Ändern des Modus führt. Bei stärkerem Druck erhöht sich jedoch die Reibung zwischen Finger und Touchscreen. Außerdem ist es schwierig einen konstanten Druck während der Operation beizubehalten. Beim ForceLock Konzept kann der Benutzer den Modus am Anfang einstellen, so dass er während der eigentlich Ziehoperation nicht darauf zu achten braucht.

Andere Arbeiten befassen sich spezifischer mit Gesten. Bhandari & Lim [8] haben Gesten als eine Eingabeform für mobile Geräte untersucht. In einer Studie sollten Teilnehmer Interaktionen zur Ausführung bestimmter Funktionen nutzen. Die Funktionen, die genutzt wurden, waren das Aufnehmen eines Fotos, und der Umgang mit der Bildverarbeitungsfunktion des Gerätes. Die Teilnehmer sagten, dass die Nutzung der Kamerafunktion eines mobilen Telefons nicht einfach sei, und sie deshalb gegenüber dieser abgeneigt waren.

In der Untersuchung mussten verschiedene Aufgaben mit dem Mobilgerät durchgeführt werden, dabei gab es 21 vordefinierte Gesten. Falls die Teilnehmer aber der Meinung waren, dass keine dieser Gesten zur Aufgabe passt, durften sie eigene "single touch" Gesten erfinden. Einige der vordefinierten Gesten waren die Zeichnung von geometrischen Figuren, das Zeichen "X" oder das Berühren des Bildschirmes mit unterschiedlichem Druck.

Die Ergebnisse der Arbeit wurden nach der Aufgabenstellung sortiert. Die benutzten Gesten von einigen Aufgabenstellungen werden hier kurz erwähnt: Bei den Aufgabenstellungen "Foto schießen" wurde der Bildschirm angetippt. Laut den Teilnehmer war dies die einfachste und intuitivste Geste.

Bei der Aufgabe "Bild löschen" haben einige der Teilnehmer ein X gezeichnet. Andere wiederum haben das Bild vom Bildschirm weggebürstet. Für die Aufgabenstellungen "nächstes Bild" und "vorheriges Bild" wurde der Bildschirm nach rechts und links bzw. nach oben und unten gescrollt.

Bhandari & Lim haben in ihrer Analyse herausgefunden, dass die Teilnehmer durch Gesten mehr angesprochen wurden, als durch "key-click" Formen der Interaktion. Durch die Gesten konnten die Teilnehmer Beispiele aus der realen Welt mit der Aufgabenstellung verbinden. Kommentare der Teilnehmer schlossen Metaphern ein, nach denen beispielsweise das X für etwas falsches steht, und so zum Löschen eines Bildes geeignet sei. Die Gesten, die von den Teilnehmern erfunden

wurden, halfen dabei die Vorstellung der Daten in den Köpfen der Teilnehmer zu verstehen. Die Teilnehmer meinten, dass sie nicht viel Zeit zum Verstehen der Eingabe verbringen würden.

Bhandari & Lim kamen zu dem Schluss, dass Gesten die Interaktion einfacher, effizienter und unterhaltsamer machen. Einiger der Teilnehmer hatten bei einigen Gesten großen Spaß und wiederholten die Geste immer und immer wieder. Außerdem wurde herausgefunden, dass Gesten den Benutzer direkt mit den genutzten Daten interagieren lassen.

Fuccella et al. haben ein System zur Textbearbeitung mithilfe von Gesten entwickelt [14]. Gesten werden dabei benutzt um verschiedene Aufgaben wie beispielsweise Kopieren und Einfügen von Text, Bewegung des Cursors und andere zu verrichten. Die Gesten wurden dabei im Bereich der Soft Tastatur gemacht. In einem Versuch wollten Fuccella et al. herausfinden, ob die Textbearbeitung durch Gesten angenehmer und effizienter wird.

Sie kamen zu dem Ergebnis, dass die Performance, abhängig von der Schriftgröße, um 13% -24% besser ist, als das Vergleichssystem auf dem Samsung Galaxy II mit Android 2.3.5. Auch die subjektive Rückmeldung der Teilnehmer war positiv. Da die beiden verglichenen Methoden verschiedene Bereiche ( einmal im Textfeld, und einmal über der Soft Tastatur) kann man sie auch zusammen benutzen.

Eine wichtige Eigenschaft bei Touchscreens ist natürlich deren Größe. Doch selbst bei den größeren Modellen kann es manchmal schwer sein ein bestimmtes Objekt, beispielsweise einen Hyperlink in einem Browserfenster, auszuwählen. Hinzu kommen noch das fat finger Problem, also dass die Berührungsfläche des Fingers vielfach grösser ist als ein Pixel auf dem Display, und das occlusion Problem, d.h. das das Ziel kurz vor dem Berühren mit dem Finger, von diesem verdeckt wird[41]. Gesture Avatar ist ein von Lü und Li präsentierte Interaktionsform die besonders bei Geräten mit kleinem Touchscreen sinnvoll ist[30]. Mit Gesture Avatar kann man beispielsweise die Form eines kleinen Widgets auf dem Touchscreen nachzeichnen. Es wird nach dem der eingegebenen Form passenden Widget gesucht, und ein Avatar(ein grösseres rechteckiges Feld) auf dem Touchscreen angezeigt. Anstatt mit dem kleinen Widget zu interagieren kann man nun den Avatar dafür benutzen. Eine andere Einsatzmöglichkeit ist es den Anfangsbuchstaben eines Links nachzuzeichnen. Gesture Avatar sucht nach dem nächstpassenden Objekt, und erzeugt wieder einen Avatar. Nun kann man anstatt des Links den Avatar antippen. Wenn man außerhalb des Avatarbereichs eine Linie in eine bestimmte Richtung zeichnet, wird der Avatar auf ein passendes Objekt in dieser Richtung gesetzt.

Lü und Li haben Gesture Avatar mit einer Shift[40] genannten Interaktionsform verglichen. Es wurde dabei die Zielerfassung untersucht. Gesture Avatar erzielte bei Zielen unterschiedlicher Größe eine viel niedrigere Fehlerrate. Außerdem war es bei kleinen Zielen signifikant schneller. Der Einsatz von Gesture Avatar während des Gehens hatte keinen bedeutenden Einfluss auf die Performanz. Gesture Avatar ist also auch für den mobilen Einsatz geeignet.

### 2.1.2 Sensoren

Aktuelle Smartphones haben eine Vielzahl von Sensoren. Sensoren messen Daten, die für verschiedene Zwecke eingesetzt werden können. Mit Hilfe der Bewegungsdaten kann man die Bewegung des Gerätes erfassen, und diese für Interaktion nutzen. Auch die Daten der anderen Sensoren lassen sich für solche Zwecke nutzen.

#### 2.1.2.1 Bewegungs- und Beschleunigungssensor

Die meisten Smartphones haben einen Beschleunigungssensor, und viele haben ein Gyroskop. Es gibt außerdem noch den Gravitätssensor, den Sensor für Lineare Beschleunigung und den Sensor für Rotationsvektoren. Diese letzten drei Sensoren können hardwarebasiert oder softwarebasiert sein. Falls sie softwarebasiert sind, bekommen sie ihre Daten von einem oder mehreren Hardwaresensoren<sup>1</sup>.

Wie auch bei zweidimensionaler Eingabe mithilfe eines Touchscreens, gibt es auch bei der dreidimensionalen Eingabe Gesten. Gesten sind dabei bestimmte Bewegungen die mit dem Gerät ausgeführt werden, beispielsweise das Zeichnen eines Buchstabens in der Luft. Bei dreidimensionalen Gesten kommen hauptsächlich der Beschleunigungssensor und das Gyroskop zum Einsatz.

Smartphones können zwar dank ihrer Sensoren dreidimensionale Gesten erkennen, doch welche Geste am Besten für bestimmte Aufgaben geeignet ist, ist unklar. Ruiz et al. haben in ihrer Arbeit versucht eine Menge von Gesten für die Best Practice zu finden[34]. Dazu wurden Teilnehmer aufgefordert für häufig genutzte Aufgaben, wie beispielsweise das Annehmen eines Anrufes, eine Geste anzugeben. Außerdem wurde mithilfe der Sensordaten und durch Videoaufzeichnungen, welche die Teilnehmer bei der Ausführung der Gesten zeigen, eine Taxonomie der Gesten erstellt. Die Ergebnisse dieser Arbeit helfen vor allem zwei Gruppen: Erstens können Entwickler durch die Klassifizierung der Gesten und durch die von Benutzern ermittelte Menge von Gesten, natürlichere Gesten entwickeln. Außerdem können sie Gesten besser auf häufig mit dem Gerät ausgeführte Aufgaben abbilden. Zweitens bekommen Hersteller von Smartphones und Entwickler von deren Betriebssystemen eine Art Anleitung, die ihnen beim Design der Sensoren und beim Erstellen ihrer Toolkits (welche Gesten sollten erkannt, und durch Apps nutzbar sein) hilft.

Den Teilnehmern wurden 19 Aufgaben vorgelegt, und sie sollten zu diesen eine Geste entwickeln und ausführen. Damit die Teilnehmer nicht durch Faktoren wie “Welche Bewegungen werden vom Smartphone erkannt?” oder “Wie sieht die aktuelle Sensortechnologie aus?” beeinflusst wurden, wurde ihnen gesagt dass sie das Smartphone als eine Art magischen Ziegelstein, welcher in der Lage ist alle

---

<sup>1</sup>[http://developer.android.com/guide/topics/sensors/sensors\\_motion.html](http://developer.android.com/guide/topics/sensors/sensors_motion.html)

ausgeföhrten Gesten zu erkennen, ansehen sollen.

Die ausgewählten Aufgaben wurden zuerst in die Bereiche Aktion und Navigation, und dann jeweils weiter in System/Telefon und Anwendung unterteilt. Die Aufgaben waren folgende: Anrufaufgaben (Annahme, Auflegen, Abweisen, Anrufen), eine Sprachsuche, Aufruf des Homescreens, vorheriges und nächstes Item (vertikal), vorheriges und nächstes Item (horizontal), hineinzoomen, herauszoomen und den Bildschirminhalt in vier Richtungen verschieben. Zu jeder entworfenen Geste mussten die Teilnehmer eine siebenpunktige Likert Skala ausfüllen. Die Aussagen waren dabei: "Die Geste die ich entworfen habe passt zu der Aufgabe" und "Die Geste ist einfach auszuführen". Außerdem wurden die Teilnehmer gefragt, wie oft sie die von ihnen entworfenen Gesten einsetzen würden. Die sechspunkte Skala hatte Werte von sehr häufig bis nie. Zum Schluss mussten die Teilnehmer noch selbst Aufgaben und zugehörige Gesten vorschlagen. Dadurch sollte herausgefunden werden, ob die vorgeschlagenen Aufgaben eine genügende Abdeckung mit der Realität aufweisen.

Die aus den Teilnehmerkommentaren extrahierten Themen waren folgende:

- die Nachahmung der normalen Nutzung, beispielsweise das Annehmen eines Anrufes durch das Halten des Smartphones ans Ohr
- eine Metapher für die Realität, beispielsweise das Umdrehen des Smartphones, so dass es mit dem Bildschirm nach unten liegt. Dies erinnert an das Auflegen eines alten Telefons.
- natürliche und konsistente Abbildung der Gesten, wenn beispielsweise die entgegengesetzte Bewegung die entgegengesetzte Auswirkung hat (vorheriges Item, nächstes Item usw.)
- der Wunsch nach einer Rückmeldung für die ausgeführte Geste

Beispiele aus der ermittelten Menge von Gesten sind das Schütteln des Gerätes zum Abweisen eines Anrufes, das Beantworten oder Tätigen eines Anrufes durch das Halten des Gerätes ans Ohr und die Auswahl der vorherigen oder nächsten Seite durch das Bewegen des Gerätes nach links, bzw. rechts.

Dicke et al. haben ein Interface vorgeschlagen, welches Gesten und räumliches Audio benutzt[10]. Es hat den Vorteil, dass der Blick nicht von anderen wichtigen Aufgaben abgewendet werden muss. Ihr System Foogue wird dabei benutzt, um beispielsweise mit Verzeichnissen und Dateien auf dem Smartphone zu interagieren. Die Elemente (Dateien und Ordner) werden dabei räumlich vor einem, in einem Drittelkreis angeordnet. Durch verschiedene Gesten mit dem Smartphone können verschiedene Items ausgewählt und abgespielt werden. Man kann beispielsweise in eine bestimmte Richtung zeigen und dann das Gerät nach oben bewegen, um ein Element auszuwählen. Der Name des Elementes wird dann vorgelesen. Dicke et al. haben eine Gestensprache mit gut voneinander unterscheidbaren Gesten für dieses Interface entwickelt.

Wolf et al. haben eine Untersuchung zu solch einem Interface durchgeführt[42]. Dabei wollten sie herausfinden, wie Benutzer mit solch einem System interagieren würden. Die Teilnehmer wurden dabei nicht auf vordefinierte Gesten eingeschränkt.

Die 10 Teilnehmer benutzten insgesamt 254 Gesten, wovon 98 3D Bewegungen waren. 147 Gesten wurden auf dem Touchscreen ausgeführt, und 19 waren eine Kombination aus 3D und 2D. Nach den Ergebnissen der Untersuchung suchten die Teilnehmer Gesten basierend auf Vorwissen und der Übertragbarkeit von Erfahrungen aus anderen Bereichen aus. Wolf et al. schlugen vor eine kleine kontextbezogene Gestenmenge, die Analogien aus anderen Bereichen und umgekehrte Gesten für eine Aktion und das Rückgängigmachen jener, unterstützt, vor. Klare und eindeutige Rückmeldung für Aktionen und Informationen über mögliche Befehle ist wichtig, um Frustration zu vermeiden, und, um eine erfolgreiche Interaktion zu garantieren.

Negulescu et al. haben die kognitiven Anforderungen von 3D Gesten und 2D Taps und Gesten in zwei verschiedenen ablenkenden Szenarien untersucht[32]. In einem Szenario muss der Benutzer gehen, im anderen ist er in einer sitzenden Position, darf aber nicht auf das Display des Gerätes schauen.

Die Untersuchung hat gezeigt, dass in beiden Szenarien, die kognitive Anforderung durch 3D Gesten statistisch nicht von denen der Taps und 2D Gesten zu unterscheiden ist. Sie schlussfolgern dass 3D Gesten eine brauchbare Eingabealternative für Situationen, in denen man nicht auf das Display schauen kann, sind.

### 2.1.2.2 Positionssensoren

Dies sind Sensoren wie Orientierungssensor, Annäherungssensor und das Magnetometer. Bei Androidgeräten sind Magnetometer und Annäherungssensor hardwarebasiert. Man kann diese Sensoren gebrauchen, um die physikalische Position des Gerätes im Bezugsrahmen der Welt zu bestimmen<sup>2</sup>.

#### Annäherungssensor

Der Annäherungssensor reagiert darauf, ob ein Objekt nah oder weit vom Gerät liegt. Beim HTC Desire beispielsweise, befindet sich dieser in der linken oberen Ecke des Gerätes. Der Annäherungssensor wird vor allem beim Telefonieren benötigt. Während des Telefonierens muss das Smartphone sehr nah an das Gesicht gebracht werden. Wenn aus Versehen der Bildschirm an der falschen Stelle berührt wird kann es passieren dass der Anruf beendet wird. Der Sensor reagiert während des Telefonierens auf die Nähe zum Kopf, und der Bildschirm wird abgeschaltet.

Da die Reichweite des Sensors beschränkt ist, und der Sensor oft nur zwischen nah und fern unterscheiden kann, ist es schwierig ihn für komplexe Interaktionen zu nutzen. Einfache Einsatzmöglichkeiten wären beispielsweise das Pausieren und Fortsetzen eines abgespielten Musikstückes, das Annehmen eines Anrufes bei gleichzeitigem Einschalten des Lautsprechers falls man schmutzige Hände hat, oder man in einer Situation ist die die ganze Aufmerksamkeit verlangt.

#### Magnetometer

Die meisten der heutigen Smartphones haben ein Magnetometer eingebaut. Dieser wird unter anderem für Kompass Applikationen benötigt. Man könnte ihn aber gut für die Interaktion des Benutzers mit dem Gerät nutzen. Dazu muss man die Veränderung des gemessenen Magnetfeldes nutzen, bzw. das Feld selber beeinflussen.

In der Arbeit von Katabdar et al. [25] interagiert der User mit dem Mobilgerät in dem er einen Magneten in Form eines kleinen Stabes oder eines Ringes benutzt. Der Magnet wird im 3D-Raum bewegt. Das Mobilgerät misst die Veränderung des Magnetfeldes mit seinem Magnetometer. Damit Faktoren wie das Magnetfeld der Erde nicht die Erkennung der Gesten verhindern, wird ein Hochpassfilter benutzt. Ein Hochpassfilter lässt hohe Frequenzen passieren und dämpft tiefe Frequenzen. Die Gesten werden durch schnelle Bewegungen des Magneten erzeugt, was zu schnellen Veränderungen des Magnetfeldes um das Mobilgerät führt. Diese Form der Interaktion mit dem Mobilgerät gehört zur ADI- Gruppe, dabei steht ADI für “Around Device Interaction”. ADI Interaktionen sind bei kleinen Mobilgeräten sehr nützlich, da man den Interaktionsraum deutlich vergrößert.

Diese Form der Interaktion wird auch in anderen Arbeiten benutzt. Katabdar

---

<sup>2</sup>[http://developer.android.com/guide/topics/sensors/sensors\\_position.html](http://developer.android.com/guide/topics/sensors/sensors_position.html)

et al. nutzen sie um Ziffern einzugeben[24]. Die Ziffern werden dabei im Raum um das Gerät in Form einer Geste geschrieben. Doch nicht nur Ziffern kann man damit eingeben. Katabdar et al. haben eine MagiSign genannte Applikation für Iphones entwickelt, die die Authentifizierung durch in der Luft gezeichnete Signaturen erlauben[23]. Dazu muss die Signatur von der Applikation zuerst gelernt werden, und kann dann zur Authentifizierung genutzt werden.

In [22] wird gezeigt, wie man mit dieser Interaktionsform eine Gitarrenapplikation auf dem Smartphone bedienen kann.

Hwang et al. haben den Raum um das Smartphone gleich mit zwei Magneten beeinflusst. Der von ihnen entwickelte und Magnetic Marionette genannte Controller ist eine Silikonhülle für Smartphones, die das Gerät bis auf die Vorderseite vollständig umschließt[19]. Auf der rechten und linken Seite sind Silikonarme angebracht, an deren Enden sich jeweils ein Magnet befindet. Dieses Arme können bewegt werden um verschiedene Gesten auszuführen. Mit diesem Controller und dem in Smartphones eingebauten Magnetometer können acht verschiedene Gesten mit 99% Genauigkeit erkannt werden. Als Vorteile geben Hwang et al. die geringere Verdeckung des Touchscreens und das bessere taktile Feedback an.

### 2.1.3 Kamera

Die Kameras der Mobilgeräte werden mit der Zeit immer besser, so dass man sie zum Erkennen von Gesten einsetzen kann. Durch den Einsatz von Kameras kann man den Raum um das Mobilgerät besser nutzen. Dies ist vor allem bei kleinen Geräten nützlich, bei denen die Bedienung per Touchscreen umständlich ist, oder solchen, die gar keinen Screen haben.

Han et al. beschreiben in ihrer Arbeit [16] wie man die Bewegung des Fußes bzw. des Beines zur Steuerung des Smartphones nutzen kann. Es werden in zwei separaten Studien die Richtung und die Geschwindigkeit des Kicks untersucht. Zur Aufzeichnung der Beinbewegungen der Probanden wird die Kamera einer Microsoft Kinect benutzt. In der ersten Studie wird untersucht, wie viele verschiedene Kickrichtungen man optimalerweise für die Steuerung des Mobilgerätes nutzen kann, d.h. in wie viele Richtungen können die Probanden kicken ohne Fehler zu machen. Ein Fehler bedeutet, dass man zu weit nach rechts bzw. links gekickt hat. Das Ergebnis der Studie ist, dass man in fünf verschiedene Richtungen kicken kann ohne zu viele Fehler zu machen. Die zweite Studie beschäftigt sich mit der Geschwindigkeit des Kicks. Für jeden Probanden werden jeweils eine minimale und maximale Kickgeschwindigkeit ermittelt. Es wird nun untersucht, in wie viele Abschnitte der Bereich zwischen der minimalen und maximalen Geschwindigkeit aufgeteilt werden kann, so dass die Probanden jeden Geschwindigkeitsabschnitt treffen können. Das Ergebnis ist, dass die Probanden gut zwischen zwei Kickgeschwindigkeiten, langsam und schnell, unterscheiden können. Diese könnten dann beispielsweise zum Zoomen eingesetzt werden. Eingesetzt werden könnte diese Interaktionsform, wenn man den Touchscreen wegen schmutzigen Händen nicht berühren will, oder wenn man Handschuhe trägt.

Die Arbeit von Baldauf et al. [7] befasst sich mit der Erkennung der Fingerspitzen der Hand, um diese später zur Gestenerkennung zu nutzen. Dazu wurde von den Autoren der Prototyp einer Fingerspitzenerkennungsengine implementiert. Wenn man die Fingerspitzen erst einmal erkannt hat, kann man diese auf verschiedene Arten nutzen, um mit dem Mobilgerät zu interagieren. Eine dieser Arten wäre die Back-of-Device-Interaction. Hier werden die Fingerspitzen mithilfe der Kamera auf der Rückseite des Mobilgerätes erkannt. Nun kann man das Kamerabild der Hand als Hintergrund des Bildschirmes des Mobilgerätes einblenden. Durch Bewegen der Finger kann man nun zum Beispiel Buttons anklicken. Anstatt das Kamerabild mit den Fingern als Hintergrund einzublenden, kann man auch die Fingerspitzen als Cursor auf der Benutzeroberfläche der Applikation darstellen. Um es etwas intuitiver zu machen, könnte der Cursor auch das Aussehen der eigenen Hand haben. Ein anderer Anwendungsfall ist die Real-World Interaction. Hierbei wird mit der Kamera des Mobilgerätes auf ein bestimmtes Objekt gezeigt. Dieses Objekt wird nun mit den Fingerspitzen ausgewählt. Über eine Datenbank kann man dann Informationen über das Objekt abrufen. Bei einem anderen aufgeführten Anwendungsfall wird durch die Finger ein rein virtuelles Objekt manipuliert. Dies kann man unter Anderem in Spielen gebrauchen, um mit virtuellen Charakteren zu interagieren. Der letzte in der Arbeit erwähnte Anwendungsfall sieht vor, dass man das Gerät wie eine Kette um den Hals trägt. Dabei zeigt die Kamera des Gerätes in die Richtung des Sichtfeldes des Benutzers. Dadurch kann man die Gestenerkennung zur Interaktion mit dem Mobilgerät nutzen, ohne, dass das Gerät den User behindert. Ein Anwendungsfall wäre, die Lautstärke von abgespielter Musik zu ändern.

Vaitukaitis und Bulling haben ein System zur Erkennung von Augengesten entwickelt [38]. Der Unterschied zu anderen ähnlichen Systemen ist, dass keine zusätzliche Hardware benötigt wird. Ihr System haben sie in einer Laborstudie auf einem Laptop und einem Android Smartphone getestet.

Das System unterscheidet zwischen sechs verschiedenen Zuständen des Auges. Fünf davon sind die Blickrichtungen Links, Rechts, Oben, Unten und Mitte. Der letzte Zustand tritt ein, wenn das Auge geschlossen ist.

Vaitukaitis und Bulling haben zwei Ergebnisse betrachtet; die Echtzeit Kapazität des Systems, und die Erkennungsperformanz. Die Ergebnisse implizierten, dass der in der Untersuchung genutzte Laptop auch unter größerer Last Videoinformationen mit einer Rate von über fünf Bildern pro Sekunde bearbeiten konnte. Auf dem Smartphone konnte bei der Erkennung von Gesicht und Augen eine Rate von fünf Bildern pro Sekunde und bei der Verfolgung etwa vier Bilder pro Sekunde erreicht werden.

Die Gestenerkennungswerte für das System waren 67,3% für den Laptop und 60% für das Smartphone. Laut Vaitukaitis und Bulling demonstrieren diese Werte nicht nur die Machbarkeit, sondern unterstreichen auch das Potenzial von Augengestenerkennung, und der Nutzung dieser für Interaktionen mit mobilen Geräten.

## 2.2 Audio

Smartphones haben schon ihrer Natur wegen ein Mikrofon zur Eingabe von Sprache und Ton. Dieses kann natürlich auch zur Interaktion benutzt werden.

### 2.2.1 Sprache

Es gibt eine Menge von Applikationen für Smartphones, die auf Spracherkennung basieren. Ein Beispiel hierfür ist die bekannte Siri Software von Apple welche auf dem Iphone läuft. Ähnliche Software gibt es auch für Android Geräte. Bei diesen Programmen wird die gesprochene Frage eines Benutzers beantwortet.

Spracheingabe und -ausgabe kommt verständlicherweise vor allem Sehbehinderten Menschen zugute. Aber auch in Situationen, in denen man das Gerät nicht die Hand nehmen kann oder nicht durch das Gerät abgelenkt werden darf, ist die Spracheingabe und -ausgabe sehr nützlich. Ein Beispiel für solch eine Situation wäre das Autofahren. Spracherkennung ist also sinnvoll, hat aber einen wichtigen Nachteil. Die Erkennung der Wörter ist nicht perfekt. Ein Spracherkennungssoftware muss also eine gute Fehlerkorrektur bieten.

Frühere Untersuchungen haben gezeigt, dass Benutzer Schwierigkeiten bei der Verbesserung von Erkennungsfehlern haben, wenn sie dafür nur die Sprache einsetzen[20]. Vertanen und Kristensson haben ein Spracherkennungssystem für mobile Geräte mit Touchscreen vorgestellt[39]. Für die Fehlererkorrektur wurde der Touchscreen genutzt. Das Parakeet genannte System zeigt den erkannten Text in einer Zeile auf dem Touchscreen des Gerätes an. Zu jedem Wort werden mögliche Alternativen angezeigt, und das erkannte Wort lässt sich durch eine dieser Alternativen ersetzen, oder, falls das erkannte Wort überflüssig ist, auch löschen. In einem durchgeführten Experiment kamen Teilnehmer in einer sitzenden Position auf 18 Wörter pro Minute, und während des Gehens auf 13 Wörter pro Minute.

Feld et al. haben sich näher mit der nachträglichen Korrektur von erkannter Sprache befasst[12]. In ihrem Experiment haben sie zwei verschiedene Spracherkennungssysteme und drei verschiedene Methoden zur nachträglichen Korrektur untersucht. Diese waren alternative Sätze, Sprachmodell, und lexikografische und phonologische Ähnlichkeiten. Es wurden die Wortfehlerrate und die Interaktionsschritte gemessen. Es wurden bei einem der Spracherkennungssysteme die Methoden alternative Sätze und Sprachmodell verglichen. Die Methode alternative Sätze wurde unterteilt in die Methoden “Direkt angewendete Satzalternativen”(DASA) und “Wortalternativen basierend auf Satzalternativen”(WASA). Wenn die niedrigste Zahl an Korrekturalternativen angeboten wurde, war die absolute Verbesserung bei der Wortfehlerrate bei DASA 4,8%, bei WASA 5.6%, und bei beim Sprachmodell 1,8%. Bei der höchsten Zahl für Korrekturalternativen waren die Werte 9,4 sowie 20,2 und 3,7. Der beste Wert für das Sprachmodell war also 3,7%. Für diesen Wert musste der Benutzer aber 10 Interaktionsschritte machen. Wenn aber situationsbedingt nur 2,6 Interaktionsschritte ausgeführt werden können sinkt der Wert von 3,7% auf 2,6%.

Eine Anwendung von Spracheingabe und Sprachausgabe haben Shirali-Shahreza

et al. vorgestellt. In ihrer Arbeit [36] haben sie zwei neue Arten von Captchas vorgeschlagen, HearSay und SeeSay. Captcha steht dabei für Completely Automatic Public Turing Test to Tell Computer and Human Apart. Captchas werden häufig bei Registrierungen im Internet oder auch bei der Eingabe von Einträgen in Foren verwendet. Dabei soll das Captcha Angriffe durch Bots verhindern. Konventionelle Captchas zeigen dem User ein Bild an, in dem ein Text steht. Dieser Text sollte für einen Computer nicht zu erkennen sein, der Mensch sollte damit keine Probleme haben. Der Benutzer muss dann den gesehenen Text eintippen. Shirali-Shahreza et al. nennen diese Methode SeeType. Für Blinde User gibt es sogenannte Audiocaptchas, bei diesen wird der Text nicht als Bild angezeigt, sondern zusammen mit Hintergrundrauschen als Audio ausgegeben. Die Antwort muss auch hier eingetippt werden.

Bei der vorgeschlagenen Methode SeeSay muss das gesehene Captcha ausgesprochen werden, beim HearSay wird das Captcha als Audio präsentiert, und der User muss die Antwort wieder aussprechen. SeeSay könnte beispielsweise beim Autofahren sinnvoll sein, wenn man also nicht die Möglichkeit hat zu tippen. HearSay könnte für blinde Benutzer einfacher sein, als die Antwort einzutippen. Shirali-Shahreza et al. wollten zwei Hypothesen testen, erstens, dass SeeSay Captchas schneller zu lösen sind als andere Arten von Captchas, und zweitens, dass die Methode SeeSay den anderen Methoden, insbesondere SeeType, vorgezogen wird. Dazu mussten Probanden Captchas vom Typ SeeType, HearType, SeeSay und HearSay lösen. Diese Captchas mussten während des Gehens oder im Sitzen gelöst werden.

Es wurde auf drei Punkte geachtet. Diese wären die Erfolgsrate, die Zeit, die zum Lösen des Captchas gebraucht wurde, und Zählungen, die die Eingabe betrafen, beispielsweise die Anzahl der Korrekturen beim Tippen, oder wie oft das Audiosample angehört wurde.

Die beste Erfolgsrate hatte die Methode SeeType, die niedrigste HearSay. Für die Dauer wurde ermittelt, dass SeeSay beim Sitzen die schnellste Methode darstellt, beim Gehen ist es gleichauf mit HearSay.

Die in der Arbeit ermittelten Ergebnisse zeigten, dass die Probanden das Aussprechen der Antworten dem Tippen vorzogen, doch hatten sie Bedenken bezüglich der Genauigkeit und der sozialen Akzeptanz des Redens mit einem automatischen System. Außerdem wurde gezeigt, dass SeeSay bequemer als aktuell genutzte Captcha Methoden ist und, dass HearSay zugänglicher als aktuelle Audiomethoden sein könnten.

### 2.2.2 Geräusche

Geräusche unterscheiden sich von Sprache dadurch, dass sie nicht aus Wörtern bestehen. Daher ist es natürlich weniger sinnvoll sie bei Aufgaben im Mobilbereich einzusetzen. Wenn ein Benutzer beispielsweise eine Datei öffnen will, ist es wahrscheinlich einfacher sich den ausgesprochenen Befehl “öffne” zu merken, als ein Geräusch dafür zu machen. Die bekannte Bedeutung des Wortes öffne

macht es einfacher diese mit einer bestimmten Aufgabe zu assoziieren. Doch es gibt Einsatzgebiete für Geräusche. Beispiele wären Benachrichtigungen für bestimmte Ereignisse wie der Eingang einer SMS oder auch das Erreichen einer bestimmten Uhrzeit(Alarm). Geräusche müssen in diesen Fällen keine weiteren Informationen beinhalten; nur das Auftreten vermittelt die Information. Baharin und Mühlberger nennen diese Interaktionen atomic interactions, und solche die spezifisch auf Geräuschen beruhen sonic atomic interactions[6]. Doch natürlich können Geräusche auch bestimmte Informationen übertragen. Das Bellen eines Hundes informiert nicht nur darüber, dass etwas in der Nähe ist, sondern auch dass es sich um einen Hund handelt. Diese Art von Information kann von lokationsbasierten Spielen genutzt werden.

Diese Spiele erfreuen sich unter Smartphone Benutzern wachsender Beliebtheit. Bei diesen Spielen wird eine virtuelle Welt auf die physikalische Welt abgebildet, d.h. bewegt man sich in der realen Welt, so bewegt man sich auch in der virtuellen Welt. Dies wird durch GPS ermöglicht. Kurczak et al. haben vorgeschlagen für diese Art von Spielen Geräusche zu nutzen[28]. Das Problem dieser Spiele ist nämlich, dass die virtuelle Welt auf dem Display des Smartphones angezeigt wird. Um sich in der virtuellen Welt fortzubewegen, muss man sich auch in der realen Welt bewegen. Da man aber durch die Anzeige der virtuellen Welt auf dem Display abgelenkt ist, kann es passieren dass man gegen ein Hindernis stößt, oder schlimmer, in einen Unfall verwickelt wird. Der Vorschlag von Kurczak et al. ist es, zur Darstellung der virtuellen Welt nicht das Display des Smartphones, sonder Umgebungsgeräusche zu benutzen. So können die Benutzer ihre visuelle Aufmerksamkeit auf die reale Welt richten, während sie über Umgebungsgeräusche die virtuelle Welt wahrnehmen. Auch könnte auf diese Weise das Empfinden, wirklich in der virtuellen Welt zu sein, steigen.

Für diese Art von Spiel muss der Spieler ein Stereo Kopfhörersystem nutzen. Man kann dann beispielsweise einen virtuellen Wasserfall durch die Geräusche in der realen Welt ausfindig machen. Ist der virtuelle Wasserfall rechts, hört man Geräusche aus der rechten Kopfhörermuschel, kommt man dem Wasserfall näher, wird das Geräusch lauter. Wenn man noch zusätzlich ein System zum Ermitteln der Kopfbewegungen einsetzt, werden die Umgebungsgeräusche noch realistischer. Kurczak et al. haben ein lokationsgestütztes Spiel namens Growl Patrol entwickelt, bei dem es darum geht virtuelle Tiere einzufangen. Um die Darstellung der virtuellen Welt auf dem Display, oder durch Umgebungsgeräusche zu vergleichen wurde das Spiel von Teilnehmern auf zwei Arten gespielt. Bei beiden wurde die Realität durch eine virtuelle Welt auf einem großen Display vor dem Spieler dargestellt. Anstatt die Spielwelt auf dem Display eines Smartphones zu betrachten, konnte man diese auf Displays, die seitlich vom Spieler angebracht waren, sehen. Falls man aber die Variante mit Umgebungsgeräuschen testete, hatte man einen Stereo Kopfhörer und Instrumente für die Ermittlung der Kopfbewegung auf.

Die Ergebnisse zeigten, dass durch die Spielweise mit Umgebungsgeräuschen die Präsenz des Spielers in der virtuellen Welt erhöht wird. Außerdem wird die Sicherheit im Gegensatz zu der visuellen Variante erhöht. Wie schon erwartet, war der Erfolg der Spieler, der durch einen Punktestand ermittelt wurde, geringer.

### 2.3 Interaktion und Merkfähigkeit

Bis jetzt wurde nur über verschiedene Interaktionsformen gesprochen. Nun wird Merkfähigkeit im Allgemeinen und der Einfluss der Interaktion auf die Merkfähigkeit betrachtet.

Das Gedächtnis ist in drei Bereiche unterteilt, dem Sensorischen Gedächtnis, dem Kurzzeitgedächtnis und dem Langzeitgedächtnis. Im Sensorgedächtnis bleiben Informationen nur einige hundert Millisekunden. Das Kurzzeitgedächtnis kann Informationen für einige Sekunden speichern [5]. Das Kurzzeitgedächtnis hat außerdem nur eine begrenzte Kapazität von  $7 \pm 2$  Informationseinheiten[31]. Wenn man sich wirklich an etwas erinnern will, muss es im Langzeitgedächtnis aufbewahrt werden.[4]

Goddon und Baddeley haben mit einem Experiment gezeigt, dass auch der Kontext beim Lernen wichtig ist. In diesem Experiment von 1971 mussten Taucher Wörter an Land oder unter Wasser lernen. Später wurden sie auch an beiden dieser Orte befragt. Wenn der Ort des Lernens und der Ort des Abrufens identisch war, konnten die Taucher eine größere Anzahl von Wörtern wiedergeben [15].

In einem Experiment von Smith mussten Teilnehmer in einem Raum lernen und in einem andern Raum wiedergeben. Die Ergebnisse waren wie erwartet schlechter als bei der Vergleichsgruppe die im gleichen Raum wiedergeben musste in dem sie auch gelernt hatte. Wenn aber die erste Gruppe sich nur den Lernraum vorstellte, reichte dies aus um die Nachteile zur anderen Gruppe wettzumachen[37].

Nun könnten auch Dinge wie Interaktion zu solch einem Kontext gehören. Das würde bedeuten, dass man Dinge, die man mit einer Interaktion gelernt hat, leichter wiedergeben kann, wenn man diese Interaktion wieder ausführt. Man müsste sich die Interaktion sogar nur vorstellen.

Alt et al. haben die kognitiven Effekte von Interaktion, speziell bei der Interaktion mit Public Displays untersucht[2]. Für die Interaktivität des Inhaltes wurden drei Unterscheidungen gemacht, dieser kann gar nicht, teilweise, oder vollständig interaktiv sein. Die Interaktion ist entweder direkt (Berühren des Touchscreens), oder indirekt (Gesten, die vor dem Display ausgeführt werden).

Alt et al. haben ein Spiel entwickelt, in welchem man Blasen zerplatzen lassen muss. Dies konnte man auf direkte Weise, indem man den Touchscreen des Displays benutzte, und auf indirekte Weise, indem man die am Public Display installierte Microsoft Kinect nutzte, machen. Die Silhouette eines vor dem Displays stehenden Teilnehmers wurde auf dem Spielfeld abgebildet, und man konnte die Blasen mit den Händen zerplatzen lassen. Zu überbringende Botschaften konnten als Hintergrund eingebracht werden, oder in den Blasen enthalten sein. Es wurde zwischen einer interaktiven Bedingung(Spieler konnte mit dem Spiel interagieren) und einer nicht interaktiven Bedingung(es wurde ein vorheriges Spiel aufgenommen, und abgespielt) unterschieden.

Es wurde eine Studie durchgeführt, in der die Teilnehmer in der Nähe des Public Display mit dem Spiel auf etwas warten mussten. Sie wussten also nicht, dass das Spiel Teil einer Studie war.

In einem zweiten Teil der Studie wurde das Public Display im Eingangsbereich eines Universitätsgebäudes aufgestellt, so dass Passanten damit interagieren konnten.

Als erstes wurden die Ergebnisse der interaktiven und der nicht interaktiven Bedingung aus dem ersten Teil der Studie verglichen. Mit den Ergebnissen haben Alt et al. gezeigt, dass sich die Interaktion signifikant auf das freie Erinnern (Abrufen einer Information aus dem Gedächtnis) nicht aber auf das Wiedererkennen (Erinnern mit Hilfe eines Reizes) auswirkt.

Zusätzlich wurden die Ergebnisse vom ersten Teil der Studie (warten, interaktiv) mit den Ergebnissen des zweiten Teils der Studie (beim passieren, interaktiv) verglichen. Es konnte aber kein signifikanter Unterschied festgestellt werden.

Diese Art von interaktiven Inhalten können beispielsweise von Ladenbesitzern eingesetzt werden, um für ihre Produkte zu werben. Die Interaktion würde dabei den Kunden helfen sich die Produkte besser zu merken.

Um Produkte vorzustellen werden oftmals Kataloge eingesetzt. Diese gibt es auch zunehmend in der Online Variante, bei der der Betrachter den Katalog mit Mausklicks vor- oder zurückblättert. Schulte et al. haben im Pageflip-Experiment untersucht, wie sich die visuellen und motorischen Reize des Umblätterns eines Katalogs auf die Merkfähigkeit auswirken[35]. Der von ihnen entwickelte Katalog kann durch Festhalten einer Ecke und nachfolgender Drag & Drop Geste umgeblättert werden. Das Umblättern wird auch visuell angezeigt. Diese Art des Umblätterns wird Pageflip genannt.

In einem Online Experiment wurde Teilnehmern vier verschiedene Arten des Umblätterns geboten, wobei jedem Teilnehmer nur eine Art zugeordnet wurde. Diese vier verschiedenen Arten waren:

- Umblättern mit einfacherem Mausklick und ohne Anzeigen des Umblättereffektes
- Umblättern mit einfacherem Mausklick und mit Anzeigen des Umblättereffektes
- Umblättern per Drag & Drop Geste und ohne Anzeigen des Umblättereffektes
- Umblättern per Drag & Drop Geste und mit Anzeigen des Umblättereffektes

Die signifikanten Ergebnisse zeigten, dass sowohl das Anzeigen des Umblättereffektes als auch das Umblättern per Drag & Drop Geste die Merkfähigkeit erhöht. Dies gilt für das Freie Erinnern und auch für das Wiedererkennen.



---

### 3 Konzept

Diese Diplomarbeit soll untersuchen, ob Interaktion uns hilft Dinge besser zu merken. Als erstes stellt sich natürlich die Frage: Womit soll interagiert werden. Dieses "Womit" wird anhand von mehreren Kriterien ausgesucht. Diese Kriterien sind folgende:

- Es sollte sich um ein elektronisches Gerät handeln.
- Es sollte mobil sein.
- Es sollte gut im Alltag zu nutzen sein.
- Es sollten verschiedene Interaktionen zur Verfügung stehen.
- Es sollte weit verbreitet sein.

Warum nehmen wir diese Kriterien zur Auswahl des Objektes mit dem interagiert wird? Durch rasante Fortschritte in der Technologie ist der Alltag immer mehr von elektronischen Geräten dominiert. Dies wird sich in der Zukunft vermutlich fortsetzen. Deshalb macht die Untersuchung anhand eines elektronischen Gerätes, die Arbeit sozusagen fortschrittlich. Denn was nützt es, wen man weiß, welches die beste Interaktionsform zum Merken eines Textes in einem Buch ist, wenn es keine Bücher mehr gibt?

Das Mobilitätskriterium wurde genommen, da das Gerät an vielen Orten einsetzbar sein sollte. Dadurch können viele Situationen in denen man sich etwas merken muss abgedeckt werden. Auch die Einsatzfähigkeit im Alltag dient diesem Punkt. Doch schreibt dieses Kriterium auch eine eingeräumt ausgeprägte Handlichkeit des Gerätes vor.

Der nächste Punkt ist sehr wichtig, da man natürlich verschiedene Interaktionsformen untersuchen und diese dann vergleichen will.

Durch die Erfüllung des letzten Kriteriums wird eine Grundlage für große Feldstudien gelegt.

Nun soll eine Auswahl anhand dieser Kriterien getroffen werden. Wenn man den ersten Punkt betrachtet, fallen einem viele Dinge wie Pc, Laptop, Tablet, Public Displays, Smartphone usw. ein.

Der zweite Punkt lässt uns die Auswahl nun eingrenzen: Der Desktop PC und das Public Displays sind nicht mobil, Laptop, Tablet und Smartphone sind dies hingegen.

Auch beim nächsten Punkt können die restlichen drei Kandidaten mithalten. Alle werden im Alltag eingesetzt. Doch der Laptop ist im Vergleich zu Tablets und Smartphones unhandlicher und wir streichen ihn deswegen.

Beim letzten Punkt haben beide Geräte wieder etwas zu bieten. Sie funktionieren mit einem Touchscreen, auf dem verschiedene Interaktionsformen wie beispielsweise mit dem Finger tippen und Wischen ausgeführt werden können. Auch haben beide Kameras und können über Sprache gesteuert werden. Doch das Smartphone

hat den Vorteil der Handlichkeit. Man kann damit besser z.B Gesten in der Luft zeichnen, es schütteln, usw. Für solche und andere bewegungsintensive Interaktionen sind Tablets eher ungeeignet. Also fällt die Wahl auf das Smartphone. Es erfüllt auch die Bedingung dass es weit verbreitet sein muss.

Hier nochmals kurzgefasst die Argumente für das Smartphone: Das Smartphone ist ein elektronisches Gerät das mobil ist, im Alltag für die verschiedensten Dinge eingesetzt wird, weit verbreitet ist, und viele Interaktionsformen zur Verfügung stellt.

Ein paar weitere mögliche Interaktionsformen wären den Bildschirm für Gesten zu nutzen, das Smartphone rotieren zu lassen, mit Hilfe des Mikrofons Geräusche oder Sprache zu erkennen, oder den Magnetsensor für die Interaktion zu nutzen. Auch kann man die Kamera oder den Annäherungssensor zur Interaktion nutzen. Wie man sieht ist das Smartphone dank seines Touchscreens und seinen vielen Sensoren das ideale Werkzeug um die Fragestellung zu beantworten. Außerdem kann man dafür geschriebene Software sehr gut verteilen, und dadurch möglichst viele Personen zur Teilnahme bewegen. Die Verteilung könnte beispielsweise über den Google Play Market laufen.

Nachdem das Medium ausgewählt wurde, muss noch bestimmt werden in welche Umgebung man die Interaktionen einbettet. Dabei ist mit Umgebung gemeint, wie der Teilnehmer zur Interaktion mit dem Smartphone motiviert wird. Man könnte beispielsweise einen interaktiven Fragebogen erstellen, bei dem man die Fragen durch Interaktion beantworten muss. Später könnten die gegebenen Antworten wieder erfragt, und dabei die Merkrate ermittelt werden. Doch das Beantworten eines Fragebogen ist eher langweilig.

Eine andere Möglichkeit wäre es unter Anderem einen Vokabeltrainer zu implementieren, bei dem der Teilnehmer mit Vokabeln interagiert. Er könnte die nächste Vokabel auf den Bildschirm wischen, Vokabeln aufdecken, usw. Hier stellt sich aber die Frage ob die Art der Vokabeln die Merkfähigkeit beeinflusst? Wie sucht man Wörter aus, die man sich gleich gut merken kann? Welchen Einfluss hat die Sprache der Vokabeln. Können sich Teilnehmer Vokabeln, die in einer ihrer Muttersprache verwandten Sprache geschrieben sind, besser merken? Man müsste auf die Herkunft der Teilnehmer Rücksicht nehmen. Wie man sieht ist diese Idee mit einigen Problemen verbunden.

Die Wahl bei dieser Arbeit fiel darauf ein Spiel zu implementieren. Der Vorteil dabei ist, dass die Motivation für den Teilnehmer schon inbegriffen ist. Der menschliche Spieltrieb ist ein guter Motivator. Ein Spiel macht Spaß, man will seine Punktzahl verbessern, usw. Als Spiel wurde ein Memory ähnliches ausgesucht. Das Spielprinzip dieses Spiels ist Teilnehmern verschiedener Herkunft und Alters entweder schon bekannt, oder sehr einfach zu erlernen. Als Motive des Spiels werden bekannte Objekte wie beispielsweise bekannte Tiere genommen. Doch auch falls man die Motive nicht kennen würde, ist es noch relativ einfach die Motive anhand ihres Aussehens zu unterscheiden. Der Unterschied von bekannten Motiven zu unbekannten Motiven wäre wahrscheinlich nicht zu groß. Sicherheitshalber

---

wurden aber trotzdem bekannte Motive (Tiere) genommen.

Trotz der Bekanntheit des Spielprinzips, hier noch eine kurze Erklärung. Es geht darum Paare von Karten mit gleichen Motiven zu finden. Die Karten sind zugedeckt, und es können maximal zwei Karten gleichzeitig geöffnet werden. Wenn gleiche Motive gefunden wurden, werden sie vom Spielfeld entfernt. Man gewinnt wenn man alle Paare gefunden hat. Das öffnen der Karten wird nun mit einer Interaktion verbunden, und man muss sich die Position und das Motiv der Karten merken.

Ein weiterer wichtiger Aspekt sind die genutzten Interaktionsformen. Wieviele und welche soll man nehmen. Damit die Anzahl der Bedingungen nicht zu groß wird, sollten nicht zu viele Interaktionsformen untersucht werden. Die Interaktionsformen sollten sich wenn möglich gut voneinander unterscheiden lassen. Es sollten nicht nur gut untersuchte Interaktionsformen genommen werden. Außerdem sollten die Interaktion von jedermann leicht durchzuführen sein. Es wurden 5 verschiedene Interaktionsformen verwendet. Diese wären:

- *Keine Interaktion*: Die Karten werden nach dem Auswählen automatisch aufgedeckt.
- *Schütteln*: Das Gerät muss zum Aufdecken der Karten in irgendeine Richtung geschüttelt werden.
- *Annäherung*: Zum Aufdecken der Karten muss der Annäherungssensor abgedeckt werden.
- *Telefon drehen*: Das Gerät muss zum Aufdecken der Karten gedreht werden ( wie eine Karte)
- *Geräusche*: Die Karten werden durch Geräusche aufgedeckt

Wieso wurden nun diese Interaktionsformen ausgewählt?

Die erste Interaktionsform, also *Keine Interaktion*, dient als Basis. Hier wird keine Interaktion genutzt sondern es muss nach dem Markieren der Karte nur kurz gewartet werden, und die Karte wird automatisch geöffnet. Man kann also diese Interaktionsform mit den anderen vergleichen, und schauen ob die Interaktion der Merkfähigkeit zugute kommt.

Bei der zweiten Interaktionsform, dem *Schütteln* des Gerätes, handelt sich um eine Interaktionsform, bei der der Benutzer sehr aktiv sein muss. Er muss das Gerät einigermaßen schnell bewegen, außerdem muss er einiges an Kraft aufwenden. Diese Punkte könnten dazu führen dass die Interaktion besser mit dem Aufdecken assoziiert wird

Die dritte Interaktionsform, also die *Annäherung*, ist eine Interaktionsform die noch nicht sehr bekannt ist. Es lohnt sich also diese zu untersuchen.

Die Interaktionsform *Telefon drehen* wurde deshalb ausgewählt, da sie sehr intuitiv ist. Das Smartphone wird nach dem Markieren der Karte zuerst umgedreht, so dass es mit der Displayseite nach unten zeigt. Man kann sich nun das Gerät als eine Karte vorstellen, die es aufzudecken gilt. Zum Zudecken der Karten muss das

Gerät wieder so gedreht werden, dass es mit der Displayseite nach unten zeigt. Der Bildschirm des Smartphones ist hier sozusagen die Vorderseite der Karte, die Rückseite der Gerätes die Rückseite der Karte. Diese Interaktionsform könnte auch mit dem realen Spiel assoziiert werden, was eventuell die Merkfähigkeit erhöht.

Die Interaktionsform *Geräusche* hebt sich von den anderen Interaktionsformen dadurch ab, dass man während des Aufdeckens der Karten keinen physischen Kontakt mit dem Gerät haben muss. Dies ist zwar auch bei der Interaktionsform *Annäherung* der Fall, doch bei dieser muss man zumindest sehr nahe an das Gerät herankommen. Außerdem bietet diese Interaktionsform eine große Vielfalt. Man kann die verschiedensten Geräusche machen, um die Karten aufzudecken. Diese Interaktionsform wäre, falls sie zu den besseren im Sinne der Merkfähigkeit gehört, eine weitere Untersuchung wert.

Ein weiterer wichtiger Punkt ist, wie gemessen wird, welche Interaktionsform am meisten der Merkfähigkeit zugute kommt. Als Möglichkeit kommen hier Daten infrage, die während des Spielens anfallen. Die Daten die gespeichert werden sind die Anzahl der Versuche die ein Teilnehmer zum Gewinnen einer Runde braucht, und die gespielte Zeit. Außerdem wird gemessen wie lange die Karten durchschnittlich offen waren, also wie lange ein Teilnehmer die Karten betrachtet hat. Dies spielt für die Merkfähigkeit eine wichtige Rolle. Diese Daten werden dann an einen Server geschickt, wo sie dann gespeichert werden. Damit die Studie anonym bleibt werden keine persönlichen Daten verwendet.

Nun ist klar, welche Daten zur Beantwortung der Fragestellung gemessen werden. Doch wer nimmt an der Untersuchung teil? In welcher Form findet die Untersuchung statt? In welcher Umgebung werden die Interaktionen ausgeführt. Diese Fragen mußten vor dem Einladen des ersten Teilnehmers beantwortet werden. Die Studie wurde in Form einer Laborstudie durchgeführt. Hierbei werden die Teilnehmer in eine kontrollierte Umgebung eingeladen. Die Bedingungen sind hier für alle Teilnehmer gleich, und man kann die Ergebnisse besser vergleichen. Ein Nachteil der Laborstudie ist, dass man die Teilnehmer nicht in ihrer alltäglichen Umgebung befragt. Die Ergebnisse können dann nicht mit hundertprozentiger Sicherheit auf die Realität übertragen werden.

Die Teilnehmer der Untersuchung kommen größtenteils aus dem universitären Umfeld, d.h. Studenten, oder Berufstätige mit einem Hochschulabschluß. Das Alter der Teilnehmer lag zwischen 20 und 30 Jahren, wobei es auch Teilnehmer gab, die etwas älter waren.

Außer der Merkfähigkeit wurde auch noch die User Experience ermittelt. Dazu wurde der UEQ Fragebogen[29] in seiner kurzen Form benutzt. Mit diesem werden Aussagen über die Attraktivität, Durchschaubarkeit, Effizienz, Originalität und Stimulation gemacht. Zwar ist das vorrangige Ziel der Arbeit den Einfluss von Interaktion auf die Merkfähigkeit zu ermitteln, doch ist es auch wichtig zu messen, ob die durchgeführten Interaktionen den Benutzern gefallen, da sie sonst nicht oder nur ungern eingesetzt werden. Die Frage nach dem Einfluss einer In-

---

teraktionsform auf die Merkfähigkeit wird sinnlos, wenn diese Interaktionsform aufgrund einer schlechten Akzeptanz bei den Benutzern nicht eingesetzt wird.



---

## 4 Implementierung

Um die Fragestellung der Arbeit durch eine Studie zu beantworten wurde ein Spiel für Smartphones entwickelt.

### 4.1 Tools

Bei der Implementierung der App wurden verschiedene Tools eingesetzt. Diese werden hier kurz vorgestellt.

*Android* ist ein Betriebssystem das vorwiegend auf Smartphones und Tablets eingesetzt wird. Es basiert auf dem Linux Kernel und wird quelloffen entwickelt. Android ist auf vielen mobilen Geräten installiert. Durch die Quelloffenheit können auch eigene Implementierungen sog. Custom Roms entwickelt und genutzt werden. Auf Android Geräten läuft die Dalvik Virtual Machine. Dieses sorgt für die Ausführung der in der Programmiersprache Java geschriebenen Apps. Dabei werden die von der Java Virtual Machine erzeugten .class Dateien zuerst in .dex Dateien (Dalvik Executable-Format) umgewandelt, welche dann auf der DVM laufen. Im Unterschied zur JVM basiert die DVM auf einer Registermaschine.

*Java* ist eine plattformunabhängige Programmiersprache. Dies wird dadurch erreicht, dass der Code durch den Java Compiler in den sogenannten Bytecode umgewandelt wird. Dieser Bytecode läuft dann auf einer virtuellen Machine der Java Virtual Machine. Diese virtuelle Maschine ist für viele Plattformen verfügbar, und sorgt für die Plattformunabhängigkeit von Java.

*Eclipse* ist eine weit verbreitete Programmierumgebung. Während Eclipse in der Vergangenheit nur zum Programmieren in der Sprache Java geeignet war, kann man es heutzutage auch für verschiedene andere Sprachen einsetzen. Eclipse lässt sich durch viele Plugins erweitern. Dadurch wird dem Programmierer einiges an Aufwand abgenommen. Bei der Programmierung von Apps für Android Geräte gibt es zum Beispiel diese Erleichterungen: Automatische Erstellung einer Xml Datei für die grafische Oberfläche einer Activity bei Erstellung der Activity und ein automatischer Eintrag in die Manifest Datei für jede erstellte Activity.

### 4.2 Architektur

Die für die Studie eingesetzte Software besteht aus einem Client, in dieser Arbeit einfach als App bezeichnet, und dem Server. Der Client hat die Aufgabe verschiedene Daten zu sammeln und diese an den Server zu schicken. Der Server hat die Aufgabe die vom Client gesendeten Daten zu empfangen und zu speichern. Da die App auf Android Geräten eingesetzt werden sollte, wurde sie in Java implementiert. Das Android Betriebssystem enthält unter anderen auch die virtuelle

Maschine Dalvik. Der Bytecode der für die Java Virtual Machine kompiliert wurde, wird in ein Format umgewandelt, das auf Dalvik läuft. Die App lässt sich in verschiedene “Bildschirme” unterteilen, den sogenannten Activities. Im Normalfall besteht jeder Bildschirm aus einer Activity. Activities können mit sogenannten Intents andere Activities starten. Die Bestandteile der grafischen Benutzeroberfläche können durch Java Code zur Laufzeit erzeugt werden, oder auch durch XML Dateien vorher definiert werden. Jede Activity gibt zu Beginn an, aus welcher XML Datei ihre Oberfläche aufgebaut ist (setContentView). Es ist auch möglich XML Dateien zur Erzeugung der ersten Oberfläche zu nutzen, diese aber später durch Code zu ändern, oder neue Objekte hinzuzufügen.

Die App besteht außerdem noch aus sogenannten Listenern. Diese warten auf bestimmte Ereignisse und reagieren auf diese in der vom Entwickler vorgegebenen Weise. Listener werden zum Beispiel eingesetzt um auf Sensordaten zu reagieren, oder auch, um die Aktion die beim Tappen eines Buttons ausgeführt werden soll, zu spezifizieren.

Als weitere Komponenten, beinhaltet die App auch verschiedene Threads. Diese werden für verschiedene Aufgaben genutzt. Eine davon wäre das Senden von Dateien.

In Abbildung 1 wird das Zusammenspiel der verschiedenen Activities der App verdeutlicht.

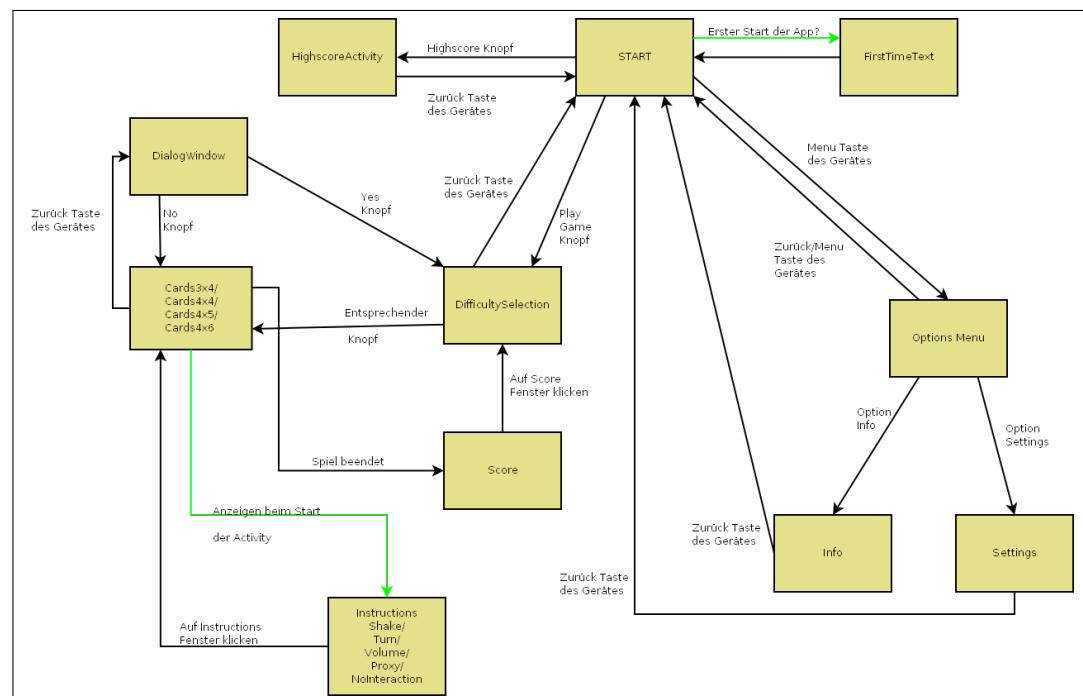


Abbildung 1: Hier sieht man die Activities der App und deren Verbindungen. Grüne Verbindungen werden automatisch und als erstes verfolgt.

### 4.2.1 Client

Der Client, eine Android App, ist eine Implementierung eines dem bekannten Memory ähnlichen Spiels. Hierbei versucht der Spieler Paare von Karten mit gleichen Motiven zu finden. Um die Karten aufzudecken werden verschiedene Interaktionen eingesetzt. Diese Interaktionen sind folgende:

- *Keine Interaktion* - Karten werden nach kurzer Verzögerung aufgedeckt
- *Schütteln* - Schütteln des Gerätes
- *Geräusche* - Öffnen der Karten durch Geräusche
- *Annäherung* - Auslösen des Annäherungssensors
- *Telefon drehen* - Aufdecken der Karten durch Umdrehen des Gerätes

Die Interaktionen sind in der Implementierung verschieden realisiert. Die Interaktion *Schütteln* und die Interaktion *Telefon drehen* nutzen den Beschleunigungssensor des Gerätes und die Interaktion *Annäherung* basiert auf den Werten des Annäherungssensors. Diese drei Interaktionsformen sind als Listener implementiert.

Die Interaktion *Geräusche* läuft als Thread und bekommt ihre Eingangsdaten über das Mikrofon des Gerätes. Auch die Interaktionsform *Keine Interaktion* nutzt einen Thread, dieser sorgt aber lediglich dafür dass die markierten Karten verzögert aufgedeckt werden.

Während der Spieler eine Runde spielt werden verschiedene Daten gesammelt. Diese wären:

- eine eindeutige Geräte ID, um mehrmaliges Spielen vom gleichen Gerät zu erkennen
- die Interaktionsform die beim Spielen genutzt wird
- die Zeit die der Benutzer zum Beenden des Levels benötigt,
- die Versuche die der Benutzer bis zum Beenden des Levels benötigt
- der Schwierigkeitsgrad den der Benutzer gewählt hat
- die Motive die beim Aufdecken der Karten angezeigt werden
- die Zeit, die vergeht, bis eine aufgedeckte Karte wieder zugeschaut wird, d.h. wie lange der Benutzer diese Karte betrachten konnte
- Ein Flag das angibt, wie diese Kopie der App verteilt wurde. Die möglichen Werte sind “TEST”- Originale App während der Entwicklung, “LAB”- Im Rahmen einer Labstudie, “MARKET”- heruntergeladen aus dem Google Play Market, oder “FRIENDS”- vom Entwickler an Freunde und Bekannte weitergegeben.

Diese Daten werden nach jeder Spielrunde in einer Datei gespeichert. Beim jedem Starten oder Beenden der App wird überprüft ob eine solche Datei vorhanden ist, und wenn ja, wird diese an den Server geschickt. Die Datei wird danach gelöscht, so dass garantiert wird, dass die gleichen Daten nicht mehrmals gesendet werden. Im dem Fall dass keine Verbindung mit dem Server aufgebaut werden kann, wird die Datei nicht gelöscht. Beim nächsten Starten oder Beenden der App wird ein neuer Sendevorschlag gestartet. Zum Senden der Daten wird eine TCP Verbindung genutzt. Dadurch wird die Datenübertragung im Gegensatz zum UDP Protokoll sicherer. Damit der Sendevorgang die App nicht blockiert, wird dafür ein eigener Thread genutzt. Wenn ein Verbindungsaufbau scheitert, oder gerade keine Datei zum Senden vorhanden ist, wir in einen catch Block gesprungen. Dadurch wird der Codeteil, der für das Senden und Löschen der Datei zuständig ist, einfach übersprungen. Der Benutzer kriegt von alledem nichts mit.

Die App besteht wie schon erwähnt, größtenteils aus Activities. Im Folgenden wird auf die einzelnen Activities näher eingegangen. Bei jeder Activity werden auch die dazugehörigen Threads, Handler und Listener erklärt. Außerdem werden am Ende noch einige andere Komponenten der App beschrieben.

### **StartActivity**

Der Startbildschirm oder auch StartActivity ist der erste Screen den der Benutzer beim Starten der App sieht. Als Oberflächenelemente hat diese Activity zwei Buttons über die man zum Schwierigkeitsauswahl-, und zum Highscorescreen kommt. Außerdem kann man durch das Drücken der Menütaste des Gerätes ein Menü im unteren Teil des Bildschirmes öffnen. Von hier aus kommt man entweder zu den Einstellungen, oder zum Infoscreen.

Während man über die Oberfläche nur die Möglichkeit hat, zu anderen Activities zu wechseln, passiert in der StartActivity einiges mehr. Es werden verschiedene Listen angelegt, über die die App später auf Bildressourcen zugreift. Diese Ressourcen sind zum Beispiel die Motive der Karten, oder die Bilder für die Kartenrückseiten. Auch wird hier das richtige Motivthema (zum Beispiel Tiere) gesetzt. Dazu wird der Wert aus einer Einstellungsdatei (welche von der Activity erzeugt wird) geladen. Im Falle, dass die Datei als Wert Random angibt, wählt die StartActivity zufällig ein Motivthema aus. Zu Beginn und beim Beenden der StartActivity (durch Drücken der Back Taste des Gerätes) wird ein SendFileThread erzeugt, welcher eine eventuell vorhandene Datei mit Daten an den Server schickt.

Beim ersten Starten der App wird eine weitere Activity gestartet. Dies ist ein Informationsfenster das den Benutzer über den Sinn der App aufklärt. Die Informationen können später über den Infobutton (im Optionsmenü) abgerufen werden. Die StartActivity hat außerdem einen Flag, der angibt in welchem Umfeld die App genutzt wird. Die möglichen Werte sind folgende: Test(für den Entwickler), LAB(in einer Labstudie), FRIENDS(Verteilung der App per Hand) und MARKET(Verteilung über den Google Play Store). Diese Werte sind als Konstanten in der Klasse Constant deklariert.

### **SendFileThread**

Mit diesem Thread wird eine TCP Verbindung mit dem Server aufgebaut. Über diese Verbindung werden dann die Spieldaten aus der auf dem Telefon gespeicherten Datei gesendet. Dazu wird der Inhalt der Datei mit Hilfe einer Scannerklasse gelesen, in einen OutputStream geschrieben, und dann mit der Methode flush() gesendet. Danach wird die auf dem Gerät gespeicherte Datei gelöscht. Dadurch wird das mehrmalige Senden der gleichen Daten verhindert.

Da der ganze Code des Threads in einem try catch Block steht, werden bei einem Fehler (keine Datei zum Senden gefunden, oder keine Verbindung zum Server) weder die Daten gesendet, noch wird die Datei gelöscht. Die StartActivity führt beim Starten und Beenden der App einen Sendeversuch aus.

### **FirstTimeText**

Diese Activity wird nur beim ersten Starten der App gestartet. Das Starten der Activity geschieht in der OnCreate Funktion der StartActivity, d.h. die Activity wird sofort beim Starten der App angezeigt. Die Activity hat nur den Zweck den Benutzer über die Studie zu informieren. Um zu gewährleisten, dass diese Information nur beim ersten Starten der App angezeigt wird, wird ein boolscher Wert mit dem Schlüssel "firstTime" aus der Preferences Datei geladen. Beim ersten Starten der App ist weder eine Preferences Datei noch der boolsche Wert vorhanden. In diesem Fall wird der Defaultwert true zurückgegeben, d.h. die App wurde zum ersten Mal gestartet. Es wird dann eine Preferences Datei angelegt, indem man einen Schlüssel namens "firstTime" mit dem Wert false darin speichert. Bei nächsten Starten der App gibt der Schlüssel "firstTime" nun den Wert false zurück und die Activity FirstTimeText wird nicht mehr gestartet.

### **DifficultySelection**

Diese Activity kann über den Startbildschirm durch das Betätigen des "Play Game" Buttons erreicht werden. Auf diesem Bildschirm hat der Benutzer die Möglichkeit aus vier verschiedenen Schwierigkeitsgraden auszuwählen, dazu gibt es vier rechteckige Buttons, die sich den gesamten Platz teilen. Außerdem wird hier, falls die Einstellung des Motivthemas auf random gesetzt ist, die Funktion der StartActivity aufgerufen die dafür sorgt, dass ein zufälliges Motivthema ausgewählt wird. Damit ist gewährleistet dass bei jedem Spielen einer Runde das Motivthema zufällig gewählt wird (und nicht nur einmal).

### **Instructions**

Hier wird wie bei der Spielfeldactivity eine Gruppe von Activities beschrieben. Die Oberfläche dieser Activities besteht aus mehreren Textfeldern und einer ImageView. Die Textfelder beschreiben wie eine bestimmte Interaktion (je nachdem welche Instructionsactivity es ist) auszuführen ist. Es wird zusätzlich noch ein Bild zur Verdeutlichung der Interaktion angezeigt. Die Activity kann durch das

Berühren des Bildschirmes geschlossen werden. Nun beginnt das Spiel und die Zeit wird gezählt.

### **Spielbildschirm**

Für das Spielen einer Runde sind eigentlich 4 Activities zuständig. Jeder Schwierigkeitsgrad hat eine eigene Activity. Diese Activities sind bis auf kleine Unterschiede identisch. Alle diese Activities implementieren das Levelinterface. Beim Starten einer dieser Activities wird sofort eine neue Activity gestartet. Diese ist ein Dialogfenster, das je nach Interaktionsform die in dieser Runde zum Aufdecken der Karten genutzt wird, eine kurze Anleitung für den Benutzer bereitstellt. Dieses "Instructions" Fenster kann durch einfaches Tappen auf den Touchscreen geschlossen werden. Damit beginnt dann eine Spielrunde, und die Zeit läuft. Abbildung 2 zeigt ein sich im Gange befindliches Spiel.

Eine Spielrunde läuft folgendermaßen ab. Der Benutzer wählt eine Karte durch berühren des Touchscreens aus. Durch die für diese Runde geltende Interaktion wird dann diese Karte aufgedeckt. Der gesamte Vorgang wird dann für eine zweite Karte wiederholt. Wenn zwei Karten geöffnet sind, können diese durch Interaktion wieder geschlossen oder vom Spielfeld entfernt werden, je nachdem ob die Motive der Karten gleich waren oder nicht. Falls die Karten geschlossen werden, wird der Zähler, der die Anzahl der Versuche anzeigt, um eins erhöht. Falls aber die Bilder der Karten gleich waren, wird zusätzlich der Zähler, der für die Anzahl der Richtigen zuständig ist, um eins erhöht. Diese beiden Werte werden zusammen mit der aktuellen Spieldauer im Titel des Spielfeldfensters angezeigt. Für das Aufdecken der Karten ist der OpenCardsHandler zuständig. Dieser ist in einer separaten Datei definiert. Als Parameter des Konstruktors wird zweimal die Spielfeldactivity, der Titlehandler, der für das Ändern der Titelzeile der Spielfeldactivity zuständig ist, und der playingTimeThread welcher die gespielte Zeit misst, übergeben.

Wenn man alle Kartenpaare gefunden hat, ist die Runde beendet. Nachdem die Runde beendet ist, wird eine neue Activity gestartet. Hier werden die Anzahl der benötigten Versuche und die benötigte Zeit angezeigt. Man kann die Spielrunde durch Drücken der zurück Taste des Gerätes jederzeit beenden. Dabei wird ein Dialogfenster angezeigt, bei dem man noch einmal das Abbrechen der Runde bestätigen muss.

Diese Activity ist die Hauptactivity der App, dementsprechend ist auch viel Funktionalität hier implementiert. Die Activity selbst und die von ihr genutzten Komponenten wie Threads, Handler und Listener sind in Abbildung 3 zu sehen. Die Karten sind jeweils als ImageView implementiert. Man kann den ImageViews nur eine Bilddatei zuordnen. Da aber auch die Rückseite der Karten angezeigt werden muss, kann man nicht beide Informationen in der Imageview speichern. Der Verweis auf die Bildresource die das Motiv der Karte darstellt ist in der Content Description der View gespeichert. Diese Zuordnung wird beim Starten der Activity gemacht. Als Bildresource wird den ImageViews am Anfang ein Kartenhintergrund zugewiesen, da die Karten zu Beginn der Spielrunde zugedeckt sind. Beim Aufdecken einer Karte wird einfach der Imageview die Bildresource, die in

der Content Description gespeichert ist, zugewiesen.

Die Karten bekommen einen Listener zugewiesen, der bestimmt wie die Karten auf Tappen reagieren. Dieser Listener besteht aus mehreren If-Anweisungen, die je nach aktuellem Zustand des Spielfeldes ausgeführt werden. Ein solcher Zustand wäre, wenn eine Karte markiert und aufgedeckt ist. Wenn jetzt eine verdeckte Karte ausgewählt wird, wird diese Karte markiert. Falls aber die aufgedeckte Karte ausgewählt wird, passiert gar nichts.

Die Funktion `writeToFile` ist für das Schreiben der Ergebnisse in eine Datei zuständig. Hier wird zuerst, falls nötig, eine neue Datei erstellt, in welche dann verschiedene Daten geschrieben werden. Diese Daten sind jeweils mit einer `#` getrennt. Die `writeScoreToFile` Funktion ist dafür zuständig den Score (Spielzeit und Anzahl der Versuche) der aktuellen Runde in eine Datei zu speichern. Außerdem wird noch der Schwierigkeitsgrad der gespielten Runde und das Datum des aktuellen Tages gespeichert. Diese Datei wird dann von der `HighscoreActivity`, die für das Anzeigen des Highscores zuständig ist, genutzt.

Die Funktion `showScoreAndFinish` ist für das Beenden der Spielrunde zuständig. Hier wird der `playingTimeThread` beendet, die Funktion `writeScoreToFile` aufgerufen und ein Intent zum Anzeigen der Activity Score gestartet. Die Spielfeld Activities haben außerdem eine Reihe von Getter- und Setterfunktionen. Diese Funktionen sind durch das `LevelInterface` vorgegeben. Genutzt wird dieses Interface unter Anderem vom `OpenCardsHandler`.

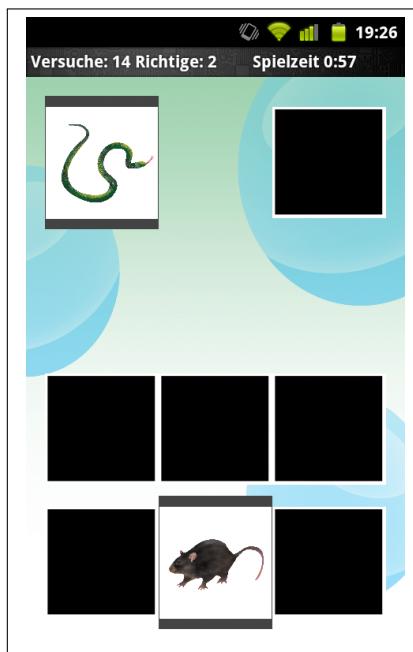


Abbildung 2: Hier ist eine laufende Spielrunde zu sehen. Es sind zwei Karten geöffnet, und mehrere Paare wurden schon gefunden (freie Stellen). In der Titelleiste sieht man Informationen wie beispielsweise die Spielzeit.

### OpenCardsHandler

Der OpenCardsHandler ist für das Aufdecken, Zudecken, und Entfernen der Karten vom Spielfeld zuständig. Dieser Handler reagiert auf eine gesendete Nachricht, wobei die Nachricht leer ist. Diese Nachricht kann von verschiedenen Stellen kommen. Die wichtigsten sind wahrscheinlich die onSensorChanged Funktionen der SensorListener. Eine andere Stelle wäre im TimerThread. Dieser ist dafür zuständig bei der Interaktionsform *Keine Interaktion* den richtigen Zeitpunkt zum Aufdecken der Karten zu bestimmen, und dann den OpenCardsHandler zu benachrichtigen.

Der OpenCardsHandler reagiert auf ein “Anstoßen”, also eine empfangene leere Nachricht, in seiner handleMessage Funktion. Als erstes wird eine Referenz auf die markierte Karte geladen. Da der OpenCardsHandler in seinem Konstruktur einen Verweis auf die Spielfeldactivity bekommen hat, kann er über diesen auf die ImageViews der Karten zugreifen

Nun kommen eine Reihe von If-Anweisungen, die je nach Spielfeldsituation ausgeführt werden. Hier eine Liste der Fälle die auftreten können, zusammen mit der Aktion des OpenCardsHandlers

- 1. Keine Karte ist markiert – mache nichts
- 2. Eine Karte markiert aber nicht aufgedeckt – decke Karte auf
- 3. Zwei Karten markiert, davon eine aufgedeckt – decke die verdeckte Karte auf
- 4. Zwei Karten aufgedeckt – decke beide Karten zu oder entferne beide Karten, je nachdem ob die Motive unterschiedlich oder gleich waren.

Im dritten Fall erhöht der OpenCardsHandler die lokale Variable, in der die Anzahl der Versuche gezählt werden. Außerdem wird eine Nachricht an den TitleHandler geschickt, um die Anzahl der Versuche im Titel der Activity, zu aktualisieren.

Falls im vierten Fall die beiden Motive gleich waren, wird die lokale Variable, die für die Anzahl der gefundenen Paare zuständig ist, um eins erhöht. Es wird wieder eine Nachricht an den TitleHandler geschickt, um den Titel entsprechend zu aktualisieren. Wenn die Karten, die auf diese Weise vom Spielfeld entfernt wurden, die letzten waren, werden die Funktionen writeToFile und showScoreAndFinish aus der Spielfeldactivity aufgerufen, da in diesem Fall die Spielrunde beendet ist. Zum Schluss werden die Werte einiger lokaler Variablen an die Variablen der Spielfeldactivity übergeben (Dies geschieht über Setter Funktionen).

### TitleHandler

Der TitleHandler ist für das Aktualisieren des Titels der Spielfeldactivity zuständig. Als Parameter des Konstruktors wird die Spielfeldactivity einmal als Activity und einmal als LevelInterface übergeben, mit der setTitle Funktion

der übergebenen Activity wird deren Titel gesetzt. Die benötigten Werte (Anzahl Versuche, Anzahl Richtige) werden mit den Getterfunktionen, die in dem übergebenen LevelInterface implementiert sind, geholt. Nur die Spielzeit wird dem TitleHandler über eine Nachricht übergeben.

### **AudioThread**

Der AudioThread wird in der SpielfeldActivity erzeugt und gestartet, falls die Interaktionsform *Geräusche* war. Der AudioThread erzeugt ein Objekt der Klasse MediaRecorder. Es werden zuerst Eigenschaften wie Quelle des Signals und Format des Outputs angegeben. Danach wird in einer while Schleife die getMaxAmplitude Funktion aufgerufen. Diese gibt die maximale Amplitude zwischen diesem und dem letzten Aufruf der Funktion zurück. Wenn dieser Wert grösser als ein bestimmter Schwellwert ist, (welcher sich in der SettingsActivity einstellen lässt) wird dem OpenCardsHandler eine Nachricht gesendet. Dieser Handler ist dann für das Aufdecken (oder Zudecken) der Karten zuständig. Die while Schleife im AudioThread läuft solange, bis ein bestimmte Variable (isInterrupted) auf true gesetzt wird. So wird beim Abbrechen oder Beenden einer Spielrunde der AudioThread beendet.

### **ShakeThread**

Dieser Thread macht nicht außer zu starten, sich für eine Sekunde schlafen zu legen, und zu Beenden. Der Thread ist nötig, damit bei der Interaktion *Schütteln*, die gleiche Schüttelbewegung nicht die Karten auf und dann gleich wieder zu deckt. Solange dieser Thread läuft (also für eine Sekunde) werden zusätzliche Schüttelbewegungen einfach ignoriert.

### **PlayingTimeThread**

Dieser Thread ist dafür zuständig die vergangene Zeit einer Runde zu messen. Dazu legt sich der Thread bei jedem Durchgang einer while Schleife für eine Sekunde schlafen. Die Variable seconds wird bei jedem Durchgang um eins erhöht. Die While Schleife hat wie beim AudioThread eine boolean Variable, um die Schleife wieder verlassen zu können.

Über den Konstruktor wird dem PlayingtimeThread der TitleHandler übergeben, welcher für die Aktualisierung des Titels der Spielfeldactivity zuständig ist. Jede Sekunde wird eine Nachricht an den TitleHandler geschickt, welcher dann den Titel ändert. Außerdem hat die Klasse noch eine Getterfunktion die die aktuelle Sekundenzahl zurückgibt.

### **ShakeListener**

Der ShakeListener wird benötigt, wenn man eine Runde mit der Interaktionsform *Schütteln* spielt. Der Listener ist eine Instanz der Klasse SensorEventListener und ist vom Typ Linear Acceleration (Beschleunigung). In der onSensorChanged

Funktion des Listeners wird überprüft ob eine Schüttelbewegung einen bestimmten Schwellenwert (einstellbar in der SettingsActivity) überschreitet, wenn ja, wird eine Nachricht an den OpenCardsHandler geschickt. Der ShakeListener ist in der Spielfeldactivity implementiert.

### ProxyListener

Dieser Listener überwacht den Annäherungssensor des Gerätes. Wenn sich ein Objekt dem Sensor annähert, ändern sich die entsprechenden Daten. Die Annäherungssensoren von vielen Geräten können meistens zwischen zwei Entfernungswerten unterscheiden, d.h. sie wissen nur ob etwas nah am Sensor ist, oder nicht. Wenn man beispielsweise mit der Hand in die Nähe des Sensors kommt, wird eine Nachricht an den OpenCardsHandler geschickt, der dann eventuell markierte Karten öffnet( oder zwei offene schliesst).

### TurnPhoneListener

Dieser Listener ist von Typ Accelerometer und reagiert auf Beschleunigungskräfte die auf das Gerät wirken. Dieser Listener nutzt die Anziehungskräfte der Erde aus um die Lage des Gerätes zu bestimmen. Wenn das Gerät waagrecht zu Erde mit der Displayseite nach unten oder oben liegt, ist der Wert den der Listener bekommt etwa 9,81 (positiv oder negativ je nachdem ob das Display nach oben oder unten zeigt). Anhand diesen Wertes kann man die Lage des Gerätes bestimmen. Wenn nun eine Karte markiert wird, und das Display erst nach unten und dann nach oben zeigt, wird die Karte geöffnet. Zum Zudecken der Karten muss das Display nach oben und dann nach unten zeigen. Der Listener reagiert schon wenn der absolute Wert kleiner als 9,81 ist, d.h. man muss das Gerät nicht drehen bis es genau waagrecht zur Erde ist

### cardsOnClickListener

Der cardsOnklickListener ist für das Markieren der Karten zuständig. Jede der Karten hat eine Instanz dieses Listeners. Der Code der hier implementiert ist sorgt für das korrekte Reagieren der Karten wenn sie angeklickt werden. Es kann gleichzeitig nur eine Karte markiert werden. Wenn man eine markierte Karte anklickt, wird die Markierung wieder entfernt.

Der cardsOnClickListener besteht aus einer Reihe von If-Statements. Zwei der auftretenden Fälle wären unter Anderem eine Karte markiert und verdeckt oder eine Karte aufgedeckt. Wenn man nun eine weitere Karte anklickt, würde deren Instanz des cardsOnClickListeners im ersten Fall das Aufdecken der Karte verhindern(im entsprechenden if- Zweig wird einfach nichts getan) und im zweiten Fall die angeklickte Karte markieren.

Dieser Listener ist immer für das Markieren der Karten zuständig, egal was die Interaktionsform ist. Nur im Fall das *Keine Interaktion* ausgewählt wurde, muss

der cardsOnlickListener mehr tun. In diesem Fall erzeugt er den TimerThread, der dann den OpenCardsHandler benachrichtigt (siehe unter Punkt TimerThread).

### TimerThread

Der TimerThread wird benötigt, wenn mit *Keine Interaktion* gespielt wird. Bei dieser “Interaktionsform” muss man nach dem Auswählen der Karte einfach eine halbe Sekunde warten, damit sich die Karte öffnet. Wenn man aber vor Ablauf der halben Sekunde die Karte noch einmal anklickt, ist sie nicht mehr markiert, und wird auch nicht mehr geöffnet. Für dieses Verhalten ist der TimerThread zuständig. Jedesmal wenn man eine Karte anklickt, wird eine Instanz des TimerThreads gestartet (siehe Listing 1). Der Thread schlafst für eine halbe Sekunde, und überprüft dann, ob außer ihm noch andere Instanzen laufen. Wenn man eine Karte zweimal schnell nacheinander anklickt, laufen zwei Instanzen des TimerThreads. Wenn die erste Instanz nach einer halben Sekunde zu ihrem Ende angekommen ist, überprüft sie ob sie die einzige Instanz ist, die gerade läuft. Diese Überprüfung geschieht, indem der Thread in einer Liste nachschaut, in der Referenzen zu allen Instanzen gespeichert sind (siehe Listing 2). Diese Liste ist in der Spielfeldactivity gespeichert, und wird dem TimerThread bei jedem Instanziieren übergeben. Wenn die erste Instanz des Threads nun sieht, dass außer ihr noch eine andere Instanz läuft, tut sie nichts und beendet sich. Wenn nun die zweite Instanz am Ende angekommen ist, sieht sie dass keine weitere Instanz des Threads läuft. Deshalb wird eine Nachricht an den OpenCardsHandler geschickt, der die Karte nun aufdeckt (oder auch nicht, je nachdem ob die Karte zuletzt markiert war).

Man kann also eine Karte nun sooft man will schnell nacheinander anklicken. Dabei werden dann jedes Mal Instanzen des TimerThreads gestartet, die aber nach einer halben Sekunde wieder beendet werden. Nur die Instanz die durch das letzte Anklicken der Karte entstanden ist, schickt eine Nachricht an den OpenCardsHandler.

```

1  if (interactionName.equals("No interaction")) {
2
3      timerList.add(new Thread(new TimerThread
4          (myHandler, timerList)));
5      timerList.get(timerList.size() - 1).start();
6  }

```

Listing 1: Hinzufügen eines TimerThreads in die timerlist mit nachfolgendem Starten des Timerthreads

```
1  for( int i=0;i<timerList . size ( ) ; i++)
2  {
3      if( timerList . get( i ) . getId ( ) ==
4          Thread . currentThread ( ) . getId ( ) )
5          { timerList . remove( i ) ; }
6      }
7
8      if( timerList . size ( ) ==0 )
9      { myHandler . sendEmptyMessage( 0 ) ; }
```

Listing 2: Test ob der aktuelle TimerThread der einzige ist. Wenn ja wird Nachricht an den OpenCardsHandler geschickt

### DialogWindow

Diese Activity wird angezeigt, wenn man während einer Spielrunde auf die Zurück Taste des Gerätes drückt. Dem Benutzer wird ein Dialogfenster angezeigt, welches einen Yes und einen No Knopf besitzt. Außerdem wird der Text “Really Cancel” angezeigt. Mit dem No-Knopf kann man das Spiel fortsetzen, der Yes-Knopf lässt das Spiel abbrechen, und man kommt zur Activity DifficultySelection zurück. Während der DialogWindow angezeigt wird, läuft die Spielzeit der Runde weiter.

### Score

Diese Activity wird am Ende der Spielrunde von der Spielfeldactivity gestartet. Es werden die gespielte Zeit und die Anzahl der Versuche übergeben. Diese beiden Werte werden dann auf dem Bildschirm angezeigt, wobei die Spielzeit zuerst in ein anschauliches Format umgewandelt wird.

### HighscoreActivity

Dieser Bildschirm ist für das Anzeigen der Ergebnisse der verschiedenen Spielrunden zuständig. Diese Activity besitzt vier Listen die jeweils 1/4 des Bildschirmes beanspruchen, und untereinander angeordnet sind. Für jeden Schwierigkeitsgrad ist eine Liste zuständig. Die Ergebnisse der Spielrunden werden aus der highscore Datei geladen, in die sie davor durch die Spielfeldactivity geschrieben wurden. Jedes Ergebnis besteht aus einem String, wobei die einzelnen Daten mit dem # Symbol getrennt sind. Die Anzahl der benötigten Versuche, die Spielzeit, welche zuerst in ein MM:SS Format konvertiert wird, und das Datum an dem die Runde gespielt wurde, werden in die Liste eingetragen. Die Einträge werden bevor sie angezeigt werden sortiert. Dabei wird zuerst aufsteigend nach der Anzahl der benötigten Versuche sortiert. Falls mehrere Ergebnisse die gleiche Anzahl von Versuchen haben, werden diese aufsteigend nach der Spieldauer sortiert.

Jeder Eintrag in einer der Listen ist durch eine Hashmap realisiert, welche die Spalten(firstColumn, secondColumn..) als Key besitzt. Die Ergebnisse werden dann als values dieser Keys gespeichert. Die Listen werden durch den Hischore-

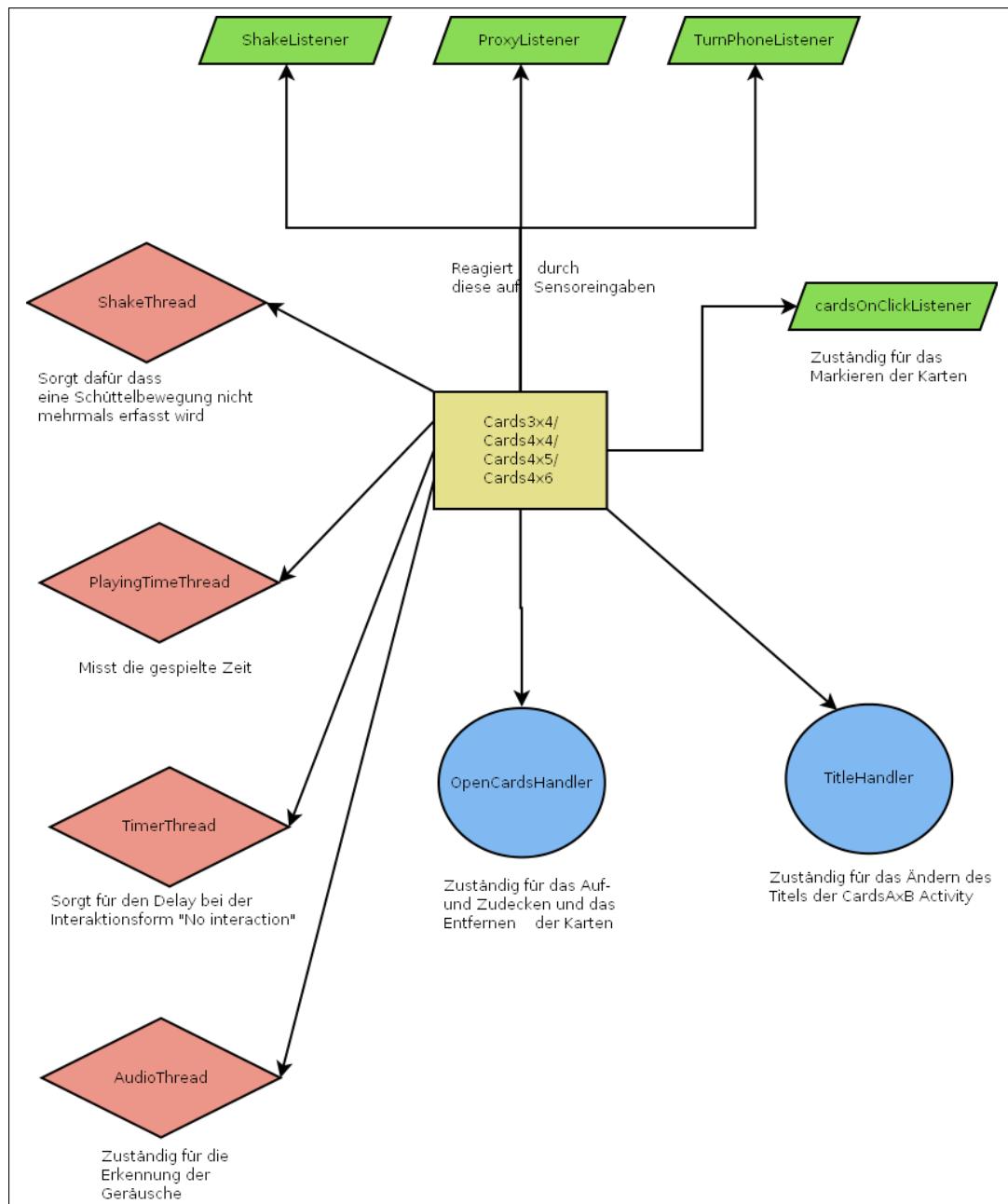


Abbildung 3: Spielfeldbildschirm (CardsAxB) und davon genutzte Komponenten. Die Activity ist in gelb gehalten, die genutzten Threads werden in rot dargestellt, Listener sind grün eingefärbt und Handler haben eine hellblaue Farbe

ListAdapter gefüllt, welcher von der Klasse BaseAdapter erbt. Abbildung 4 zeigt die HigscoreActivity.

### HighscoreListAdapter

Dieser Adapter füllt die Listen der HighscoreActivity mit den Ergebnissen der Spielrunden. Die Informationen kommen aus Hashmaps, wobei die Keys der Hashmaps als konstante Strings definiert sind. Jede Hashmap hat die Keys First, Second und Third. Die Werte dieser Keys sind die Anzahl der Versuche, die gespielte Zeit und das Datum an dem die Runde gespielt wurde. Der Adapter



Abbildung 4: Die HighscoreActivity mit Einträgen. Es sind fast nur die Listen für 3x4 und 4x6 gefüllt, da diese Spielfeldgrößen in der Studie verwendet wurden.

schreibt diese Werte in die entsprechenden TextViews der Highscorelisten. Außer dem Füllen der Highscorelisten sorgt der HighscoreListAdapter noch dafür dass die einzelnen Zeilen der Listen sowie die Listen als Ganzes nicht markiert werden können. Dies wird durch die Funktionen isEnabled und areAllItemsEnabled realisiert, wobei die beiden Funktionen den Wert false zurückgeben.

### Info

Diese Activity ist nur zum Anzeigen von verschiedenen Informationen zuständig. Hier wird der Benutzer über den Zweck der App informiert. Außerdem werden die Inhaber von vielen der genutzten Bilddateien erwähnt. Diese Bilder sind von der Seite flickr.com und stehen unter der Creative Common Lizenz. Die meisten Bilder können unter der Voraussetzung einer nicht kommerziellen Nutzung, und der Angabe des Urhebers genutzt werden. Diese Angabe wird in Form des flickr usernames und einem Link zu dem Bild gemacht. Der Anfang des Textes dieser Activity stimmt mit dem der Activity FirstTimeText überein.

### SettingsActivity

Diese Activity ist für diverse Einstellungen der App zuständig und ist in Abbildung 5 zu sehen. Die Oberfläche der Activity besteht aus vier Drop Down Menüs, und zwei Spinners. Mit den Drop Down Menüs lassen sich zu nutzende Interaktion, das zu nutzende Motivthema und das Hintergrundbild der Karten einstellen. Die ersten zwei Werte können dabei auch auf zufällig, d.h. auf Random gesetzt werden. Mit den beiden Spinners werden die Sensibilität der Interaktionsformen *Schütteln* und *Geräusche* eingestellt.

Damit die Einstellungen ein Beenden und Neustarten der App überstehen, wer-

den sie in einer Datei gespeichert. Dazu gibt es bei Android die Möglichkeit eine Preferences Datei anzulegen. Hier werden Key Value Pairs gespeichert. Die Values können dabei Zahlen oder Strings sein. Jedes Mal wenn eine Einstellung geändert wird, wird dies sofort in die Preferences Datei geschrieben. Beim Starten der Activity werden Menü- und Spinnerobjekte mit den Werten aus der Preferences Datei initialisiert. Falls noch keine Preferences Datei vorhanden war (erster Start der SettingsActivity), werden Defaultwerte zur Initialisierung genommen.

Für die Motive der Karten wurden verschiedene wohlbekannte Objekte benutzt. Die Motivthemen sind Tiere, Obst und Gemüse, Werkzeuge und Möbelstücke.

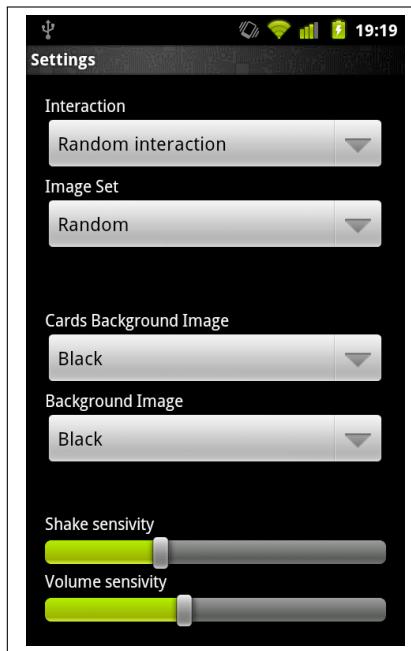


Abbildung 5: Hier können Einstellungen zur App gemacht werden.

## LevelInterface

Das Levelinterface ist eine Interface Klasse die von allen Spielfeldactivities (eine Activity pro Schwierigkeitsgrad) implementiert wird. Sie besteht grösstenteils aus Getter- und Setterfunktionen. Außerdem sind hier noch Funktionen wie writeToFile, addOpenCardsDurationList, und showScoreAndFinish definiert.

Das LevelInterface wird vom OpenCardsHandler und von TitleHandler genutzt. Dadurch können diese Handler auf Funktionalität, welche in einer der vier Spielfeldactivities implementiert ist, zugreifen, ohne zu wissen um welche Activity es sich genau handelt. Funktionen auf die der OpencardsHandler zugreift wären z.B Getter- und Setterfunktionen um die Anzahl der Versuche und die Anzahl der Richtigen zu lesen oder zu schreiben. Außerdem wird beim Ende einer Spielrunde die writeToFile Methode aufgerufen, um die Spielergebnisse in eine Datei zu schreiben. Auch greift der OpenCardsHandler auf die Funktion addOpenCardsDurationList zu, um damit die Zeit, die eine Karte aufgedeckt war, zu speichern. Diese wird dann später in die Ergebnisdatei geschrieben.

Der TitleHandler holt sich mit den Gettern des LevelInterface die Anzahl der Ver-

suche und die Anzahl der Richtigen um diese in den Titel des Spielbildschirms zu schreiben.

### Constant

In dieser Klasse sind verschiedene Konstanten deklariert. Ein Beispiel sind die Strings die dem player Flag der MainActivity als Werte dienen. Ansonsten gibt es hier noch Konstanten die bei der Erstellung der Highscorelisten benötigt werden.

### AndroidManifest

Die AndroidManifest.xml Datei beinhaltet verschiedene Informationen über die App. Es werden beispielsweise die kleinste notwendige Android Version und die Zielversion angegeben. Außerdem sind hier alle Activities der App eingetragen. Diese können auch eigene Einstellungen haben, unter Anderem dass die Activity nur im 'portrait' Modus genutzt werden kann. Dies heisst dass sich die Activity beim Drehen des Gerätes nicht mitdreht. Alle Activities der App haben diese Einstellung.

Auch Erlaubnisse, welche die App benötigt, müssen in der AndroidManifest Datei angegeben werden. Die App benötigt die Erlaubnisse INTERNET um die Daten an den Server zu schicken, und RECORD\_AUDIO um bei der Interaktionsform *Geräusche* reagieren zu könne. Die aufgenommenen Geräusche werden aber nicht gespeichert, sondern sofort verworfen.

#### 4.2.2 Server

Der Server ist dafür zuständig die Ergebnisse, die er vom Client bekommt zu empfangen und zu speichern. Die Verbindung basiert auf TCP. Das TCP Protokoll ist im Gegensatz zum UDP Protokoll ein verbindungsorientiertes, zuverlässiges Protokoll. Aus diesen Gründen wird es zur Datenübertragung genutzt.

Für jede an den Server ankommende Verbindung wird eine Datei angelegt, in welche die empfangenen Daten gespeichert werden.

Die Serverimplementierung wurde in Java geschrieben, und besteht aus zwei Klassen, der Main Klasse und der SocketThread Klasse.

### Main

In der Main Klasse wird zuerst ein Objekt der Klasse ExecutorService erzeugt indem die Methode newFixedThreadPool() aufgerufen wird. Diese Methode gibt ein Objekt der Klasse ExecutorService zurück. Der Parameter der Funktion gibt an wieviele Threads gleichzeitig genutzt werden können. Als nächstes wird ein Objekt der Klasse ServerSocket erzeugt. Der Konstruktor bekommt als Parameter den Port auf dem der Server hören soll (10923). Nun wird auf diesem Port gehört. Dazu wird die accept() Methode des ServerSocket Objekts genutzt. Wenn eine Verbindung zustande kommt, gibt die accept Methode ein Objekt der Klasse ClientSocket zurück.

Nun wird die execute() Methode des ExecutorService-Objekts aufgerufen. Als

Parameter muss ein Objekt mit der Schnittstelle Runnable übergeben werden. Das ExecutorService-Objekt kann mehrere solcher Runnable Objekte verwalten, d.h. es können mehrere Runnable Objekte gleichzeitig laufen. Wenn ein Thread des ThreadPools frei wird, wird das nächste, mit der execute() Methode in die Schlange eingereihte, Runnable Objekt abgearbeitet.

Dem execute() Methode wird nun ein Objekt der Klasse SocketThread übergeben. Diese selbst implementierte Klasse implementiert die Schnittstelle Runnable, und der Konstruktor bekommt das ClientSocket Objekt. Die accept() Methode und die execexcute() Methode stehen in einer endlosen Whileschleife, d.h. nach dem Aufruf der execute() Methode wird per accept() auf eine neue eingehende Verbindung gewartet.

## SocketThread

Hier wird zuerst ein Verzeichnis für die Dateien, in denen die ankommenden Daten gespeichert werden, angelegt. Danach wird ein Objekt der Klasse File erzeugt, und als Verzeichnis das zuvor generierte angegeben. Die ankommenden Daten werden aus dem von der Mainklasse übergebenen ClientSocket-Objekt gelesen. Diese werden als InputStream-Objekt gespeichert, welches wiederum in ein Objekt der Klasse BufferedReader gepackt wird. Nun werden die Daten mit der readLine() Methode des BufferedReader-Objekts Zeile für Zeile gelesen und in einem String gespeichert. Nach jeder Zeile wird die Escapesequenz “Neue Zeile” eingefügt.

Nun wird mit der createNewFile() Methode des vorher erstellten File-Objekts eine neue Datei auf der Festplatte angelegt, und in diese mit einem FileWriter-Objekt geschrieben.



---

## 5 Studie

Die Studie ist der zentrale Teil dieser Arbeit. Es soll in der Studie herausgefunden werden, ob man sich Information durch Interaktion besser merken kann. Es wurde dazu ein Memory ähnliches Spiel für Android Smartphones implementiert. Dieses Spiel eignet sich sehr gut für die Beantwortung der Fragestellung dieser Arbeit, da es bei dem Spiel hauptsächlich um Merkfähigkeit geht. Außerdem muss man Karten 'aufdecken', und dies kann man gut mit verschiedenen Interaktionen kombinieren.

Die Studie wurde in Form einer Laborstudie durchgeführt. Hierbei werden die Teilnehmer in eine kontrollierte Umgebung eingeladen. Die Bedingungen sind hier für alle Teilnehmer gleich, und man kann die Ergebnisse besser vergleichen. Ein Nachteil der Laborstudie ist, dass man die Teilnehmer nicht in ihrer alltäglichen Umgebung befragt. Die Ergebnisse können dann nicht mit hundertprozentiger Sicherheit auf die Realität übertragen werden. Dies bedeutet dass die Externe Validität(wie beispielsweise in [3] erklärt) in einer Laborstudie allgemein kleiner ist, als beispielsweise in einer Feldstudie, welche in der gewohnten Umgebung des Teilnehmers stattfindet.

Die Interne Validität [3] sagt aus, dass die Ergebnisse der abhängigen Variablen im großen Maße von den unabhängigen Variablen abhängen, und nicht von anderen Faktoren beeinflusst werden. In einer Feldstudie ist es natürlich schwieriger alle Faktoren zu kontrollieren, da man nicht bei jedem Teilnehmer vor Ort sein kann. Bei einer Laborstudie wird dies durch die kontrollierte Umgebung stark vereinfacht. Damit Größen wie beispielsweise die Reihenfolge von Aktionen keinen Einfluss auf das Ergebnis haben, kann die Reihenfolge für jeden Teilnehmer permutiert werden. So muss kein Teilnehmer die Aktionen in der gleichen Reihenfolge ausführen, was die interne Validität erhöht.

Die Studie wurde mit 30 Teilnehmern durchgeführt. Davon waren sechs Teilnehmer weiblich und 24 Teilnehmer männlich. Das Alter der Teilnehmer lag zwischen 20 und 39 Jahren. Der Mittelwert des Alters war 27,2 und die Standardabweichung betrug 4,49. Die Teilnehmer wurden größtenteils aus dem Universitätsumfeld rekrutiert. Die Studie wurde größtenteils in einem Büro im Institut durchgeführt. Gelegentlich wurde auch ein Laborraum dafür genutzt. Die genutzten Räume hatten die gleichen Bedingungen. Abbildung 6 zeigt einen Teilnehmer während der Studie.

Den Teilnehmern wurde zuerst erklärt, worum es in der Studie geht. Damit die Teilnehmer nicht durch das Wissen der Fragestellung beeinflusst wurden, wurde ihnen erzählt, dass es um die User Experience bei der Interaktion mit Smartphones geht. Der Studienleiter teilte ihnen mit dass es sich um ein Memoryspiel handelt. Danach wurden sie über den Ablauf der Studie informiert, welcher folgendermaßen aussah: Zuerst wurde die zu nutzende Interaktion vom Studienleiter eingestellt. Dieser erklärte den Teilnehmern nun die Interaktionsform die in dieser Interaktion zu nutzen ist. Die Teilnehmer durften danach selber etwas üben, bis sie die Interaktionsform verstanden hatten, und in der Lage waren sie einzusetzen. Als Nächstes mussten die Teilnehmer mehrere Spiele mit dieser Interaktionsform

spielen. Wenn sie fertig waren mussten sie einen Fragebogen zu dieser Interaktionsform ausfüllen. Der Fragebogen war ein UEQ(User Experience Questionnaire) Fragebogen (siehe Anhang). Am Schluss des Fragebogens waren die verschiedenen Interaktionen aufgeführt, und die Teilnehmer mussten die in der Iteration genutzte Interaktionsform ankreuzen. Diese Iteration wurde dann für jede Interaktionsform durchgeführt.

Die Reihenfolge in der die Interaktionen genutzt wurden war permutiert, und jedem Teilnehmer wurde eine andere Permutation zugewiesen. Für das Ermitteln der Permutationen wurden sogenannte Latin Squares genutzt. Durch diese Permutationen wurde die interne Validität der Studie erhöht. Die Teilnehmer mussten zu jeder Interaktionsform ein Spiel mit 24 Karten und zwei Spiele mit zwölf Karten spielen. Auch diese Reihenfolge wurde permutiert.

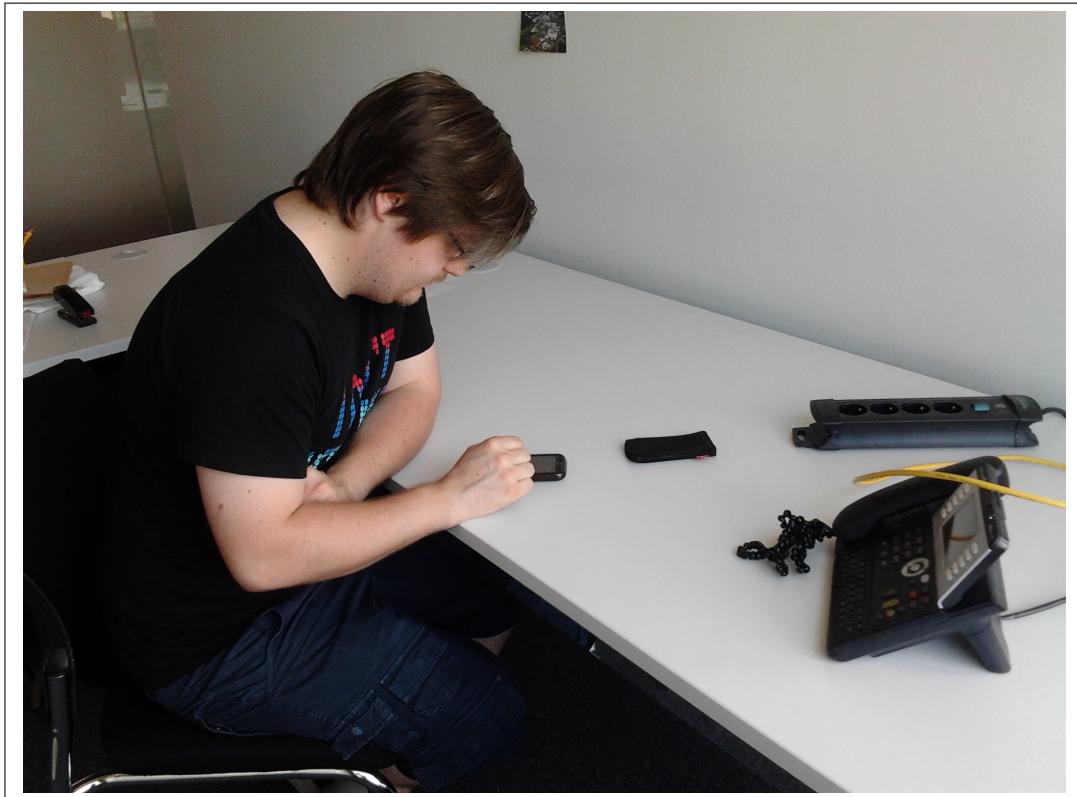


Abbildung 6: Ein Teilnehmer beim Interagieren mit dem Gerät. Die Interaktionsform ist *Keine Interaktion*

Bei den einzelnen Interaktionsformen bekamen die Teilnehmer einige Hinweise. Bei der Interaktionsform *Keine Interaktion* wurde ihnen gesagt dass sich die Karten nach einer Verzögerung von 500 ms selbst öffnen, man also nicht viel Zeit hat um die Karte wieder abzuwählen, falls man sich für eine andere Karte entscheiden sollte.

Bei der Interaktionsform *Schütteln* wurde darauf hingewiesen zwischen einzelnen Schüttelbewegungen eine sehr kurze Pause einzulegen, da bei zwei schnell hintereinanderfolgenden Schüttelbewegungen die Zweite nicht erfasst wird. Diese Pause zwischen den Schüttelbewegungen ist beabsichtigt, da es sonst passieren kann, dass

---

man durch eine Schüttelbewegung die zweite Karte aufdeckt, und beide Karten sofort wieder zudeckt, man also keine Zeit hat sich die zweite Karte einzuprägen. Bei der Interaktionsform *Geräusche* wurde darauf geachtet, dass es ruhig ist, d.h Fenster waren geschlossen, und der Studienleiter war ruhig. Es wurde den Teilnehmern vorgeschlagen kurze Geräusche zu machen, da bei langen Geräuschen, wie längeres Sprechen, die Karten auf- und dann wieder zudeckt werden können. Man hätte dann also wenig Zeit sich die Karten einzuprägen. Als Vorschläge für Geräusche wurden vom Studienleiter Fingerschnipsen, Hände klatschen und auf den Tisch klopfen vorgeschlagen.

Zur Interaktionsform *Telefon drehen* wurde den Teilnehmern gesagt, dass sie das Gerät als eine Karte ansehen sollen. Zuerst musste das Gerät mit der Displayseite nach unten gedreht werden. Nun repräsentierte das Gerät eine zudeckte Karte. Aufdecken konnte man die Karte, indem man das Gerät “aufdeckte”, also umdrehte. Wenn man zwei Karten offen hatte konnten diese wieder zudeckt werden indem man das Gerät “zudeckte”, also umdrehte.

Der Studienleiter gab bei der Interaktionsform *Annäherung* an, wo der Annäherungs-

sensor des Gerätes sei. Auch wurde den Teilnehmern erklärt, welche Probleme bei der Aktivierung des Annäherungssensor auftreten können. Wenn der Annäherungs-

sensor einmal aktiviert wurde (zum Beispiel um eine Karte aufzudecken) musste man die Hand wieder von Sensor entfernen. Falls man die Hand nahe am Sensor behielt und gleichzeitig mit der anderen Hand die nächste Karte markierte, wurde diese nicht aufgedeckt. Um diese Karte aufzudecken musste man die Hand vom Sensor weg bewegen und dann wieder nah an den Sensor bringen.

Als Vorschlag zur Bedienung des Sensors wurde zum einen das frontale Annähern der Hand an den Sensor, und zum anderen eine Wischbewegung mit der ganzen Hand über die gesamte Vorderseite des Gerätes genannt. Wenn diese Bewegung richtig ausgeführt wurde, wurde der Sensor aktiviert und die Karten öffneten/schlossen sich.



## 6 Ergebnisse der Studie

In diesem Abschnitt werden die Ergebnisse der Studie näher betrachtet. Dabei gibt es zwei Arten von Ergebnissen. Zum einen werden die automatisch an den Server übermittelten Daten betrachtet, zum anderen werden die von den Teilnehmern ausgefüllten Fragebögen ausgewertet.

### 6.1 Einfluss auf die Merkfähigkeit

Nach jeder Spielrunde wurden Spieldauer, Anzahl der Versuche, Zeit die die Karten durchschnittlich offen waren usw. an den Server geschickt.

Zuerst werden ein paar Statistiken für das Maß Anzahl der Versuche betrachtet. Dies machen wir einmal für den Schwierigkeitsgrad zwölf und einmal für den Schwierigkeitsgrad 24 Karten. Abbildung 7 zeigt die Werte für zwölf Karten.

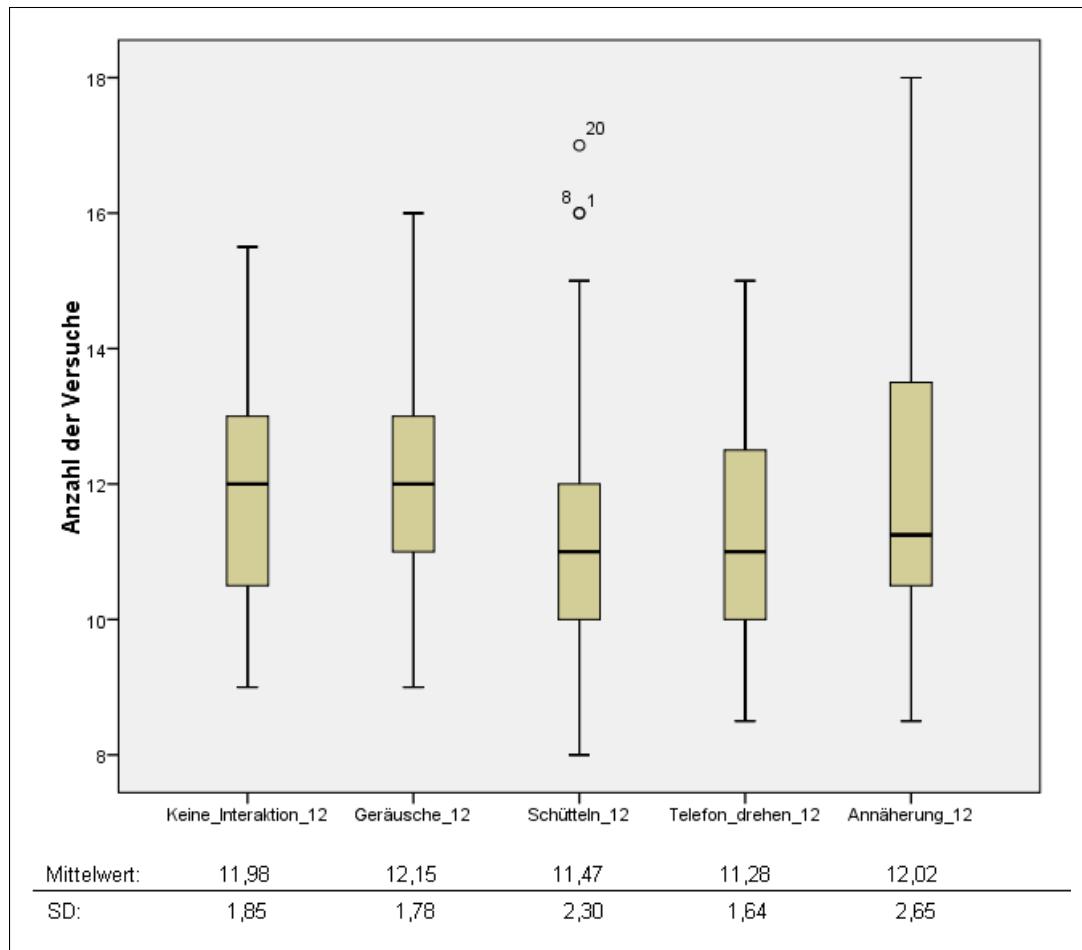


Abbildung 7: Boxplot mit zusätzlicher Angabe von Mittelwert und Standardabweichung für Anzahl der Versuche - zwölf Karten

Der Mittelwert ist bei der Interaktionsform *Geräusche* mit 12,15 am höchsten. Den zweithöchsten Wert hat die *Annäherung*, hier ist der Mittelwert 12,02. Die Interaktionsform *Keine Interaktion* ist etwas besser und liegt knapp unter 12. Die Mittelwerte von *Schütteln* und *Telefon drehen* sind mit 11,47 und 11,28 deutlich

niedriger als bei den anderen Interaktionsformen. Durch Minimum und Maximum Werte wissen wir die kleinsten und grössten Werte pro Interaktion. Aus der Differenz berechnet sich die Spannbreite. Schütteln hat mit dem Wert acht das kleinste Minimum, dies bedeutet, dass mindestens ein Teilnehmer zwei Spiele mit acht Versuchen gewonnen hat (Die angegebenen Werte sind immer Mittelwerte von zwei Spielen). Dies ist ziemlich gut da das mögliche Minimum bei sechs liegt. *Telefon drehen* hat mit einem Wert von 8,5 das nächstbeste Minimum. Die anderen Interaktionsformen haben entweder 9 oder sogar 9,5 (*Annäherung*) als Minimum.

Bei dem Maximalewerten hat die Interaktionsform *Annäherung* mit 19 die höchste. Dadurch ist die Spannbreite, also die Grösse des Bereichs in dem die Werte liegen, bei *Annäherung* am grössten. Dies führt auch zu der höchsten Standardabweichung von 2,65. *Schütteln* hat mit 17 auch einen recht hohen Maximalwert. Die anderen Interaktionsformen sind alle niedriger und haben die Werte 16 (*Geräusche*), 15,5 (*Keine Interaktion*) und 15 (*Telefon drehen*). Die Spannweiten sind bis auf *Schütteln* (9) dementsprechend deutlich niedriger als bei *Annäherung*.

Nun werden noch die Perzentile betrachtet. Diese sagen aus wie viele Werte in welchem Bereich liegen. Bei *Schütteln* und *Telefon drehen* sind 75 % der Werte kleiner oder gleich 12,25 bzw 12,5. Dies bedeutet dass nur 25 % der Werte darüber liegen. Bei den anderen Interaktionsformen sind 25 % der Werte grösser oder gleich 13 (*Geräusche*), 13,13 (*Keine Interaktion*) und 13,5 (*Annäherung*). Auch beim Median sind die Interaktionsformen *Schütteln* und *Telefon drehen* die besten, beide haben einen Wert von 11.

Zusammenfassend kann man sagen dass *Schütteln* und *Telefon drehen* die Interaktionsformen sind, bei denen die Merkfähigkeit am besten war. Beide haben einen sehr niedrigen Mittelwert, und die Standardabweichung von *Telefon drehen* ist im Vergleich zu den anderen Interaktionsformen die Beste. *Schütteln* hat zwar eine hohe Standardabweichung, aber der Median von elf sagt aus, dass dafür eher sehr kleine Werte verantwortlich sind, als sehr große. Bei der Interaktionsform *Annäherung* war die Merkfähigkeit am schlechtesten. Der Mittelwert war der mit geringem Abstand der zweithöchste und 25 Prozen der Werte lagen bei 13,5 oder darüber.

Nun werden die Werte für den Schwierigkeitsgrad 24 Karten betrachtet. Diese sind in Abbildung 8 zu sehen.

Hier hat die Interaktionsart *Annäherung* mit 29,20 den kleinsten Mittelwert. Gefolgt wird dieser von 29,83 bei *Keine Interaktion*. *Telefon drehen* hat einen Mittelwert von 31,37 und *Schütteln* und *Geräusche* haben beide mit 32,67 den höchsten Mittelwert.

Die Minimalen Werte liegen bei *Telefon drehen* und *Annäherung* bei 20, und bei *Schütteln* und *Keine Interaktion* bei 21. Nur *Geräusche* hat mit 17 einen deutlichen niedrigeren Minimalwert.

Bei den Maximalwerten liegen *Telefon drehen* und *Annäherung* mit 43 bzw. 41 wieder nahe beieinander, genauso wie *Schütteln* und *Keine Interaktion* welche die

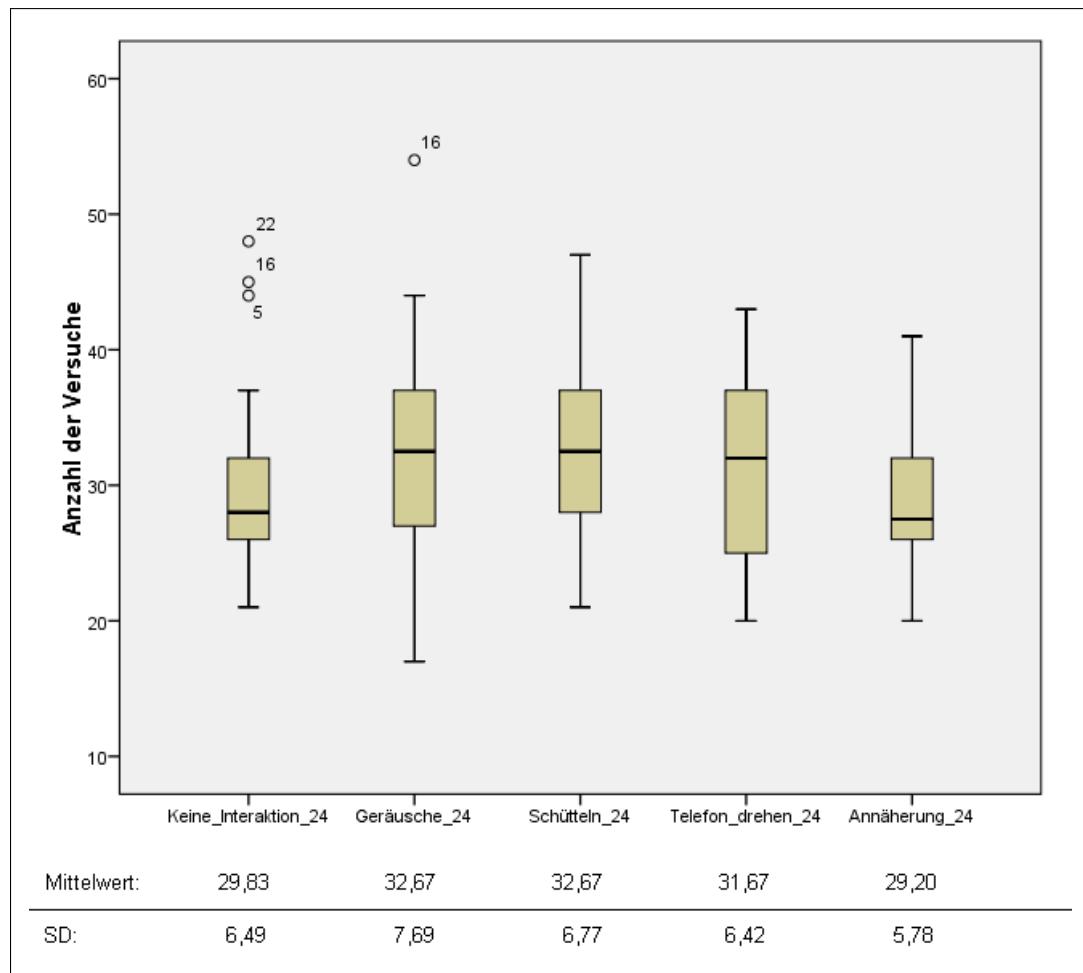


Abbildung 8: *Boxplot mit zusätzlicher Angabe von Mittelwert und Standardabweichung für Anzahl der Versuche - 24 Karten*

Werte 47 bzw. 48 haben. Die Interaktionsform *Geräusche* liegt auch hier abseits und hat einen Maximalwert von 54.

Die Spannweite ist bei *Annäherung* am niedrigsten und beträgt nur 21, *Geräusche* hat mit 37 den höchsten Wert. Die anderen Interaktionsformen haben Spannweiten von 23 (*Telefon drehen*), 26 (*Schütteln*) und 27 (*Keine Interaktion*). Die kleinsten und grössten Spannweiten spiegeln sich auch in den Standardabweichungen wieder. *Annäherung* hat mit 5,78 die niedrigste Standardabweichung und *Geräusche* mit 7,7 die höchste.

Bei den Perzentilen sehen wir dass bei *Annäherung* 25 % der Werte kleiner oder gleich 25,75 sind, und 25 % über 32,25 liegen. Ähnliche Grenzen sind nur bei *Keine Interaktion* zu sehen. Hier sind die unteren 25 % kleiner oder gleich 26 und die oberen 25 % größer oder gleich 32. Die anderen Interaktionsformen haben für die untere Grenze Werte von 25 bis 28, doch die grössten 25 % der Werte sind größer oder gleich 37.

Bei dem Schwierigkeitsgrad 24 Karten hat die Interaktionsform *Annäherung* am besten abgeschnitten, d.h. die Merkfähigkeit war hier am Größten. Der Mittelwert und der Median von *Annäherung* sind jeweils die kleinsten. Außerdem

gehören die Grenzwerte für die unteren und oberen 25% zu den niedrigsten. Als zweites folgt die Interaktionsform *Keine Interaktion*. Der Mittelwert und Median sind hier jeweils die zweitbesten, und auch die Werte für die Perzentile sind nicht schlecht. Am schlechtesten schnitten *Geräusche* und *Schütteln* ab. Sie haben beide die gleichen hohen Werte für Mittelwert und Median. Außerdem sind ihre Werte für die Perzentilen fast gleich. Der Unterschied zwischen diesen beiden Interaktionsformen ist die Standardabweichung und die Werte für Minimum, Maximum und Spannweite. Während bei *Schütteln* die Standardabweichung und die Spannweite niedriger waren, waren sie bei *Geräusche* sehr hoch. Dies kann damit zusammenhängen dass die Teilnehmer je nach genutztem Geräusch gut mit dieser Interaktionsform klarkamen oder nicht.

Zusätzlich zu diesen Statistiken wird pro Schwierigkeitsgrad noch eine einfaktorielle Varianzanalyse mit Messwiederholung durchgeführt. Der einzige Faktor ist die Interaktionsart. Als erstes muss die Normalverteilung überprüft werden. Dies kann mit einem entsprechenden Statistikprogramm gemacht werden.

Die Überprüfung zeigt dass die Daten normalverteilt sind. Da die Daten normalverteilt sind kann man nun mit der Varianzanalyse beginnen. Als erstes betrachtet man den Schwierigkeitsgrad zwölf Karten. Zuerst wird die Sphärizität überprüft. Dafür nutzt man den Mauchly Test auf Sphärizität.

Der Signifikanzwert liegt bei 0,038, dass heißt dass die Sphärizität nicht gegeben ist. Man muss also die Freiheitsgrade korrigieren. Dies macht man mit der Greenhouse-Geisser Korrektur.

Die Ergebnisse sind nicht signifikant, deshalb kann man nicht weiter analysieren.

Die gleiche Methode wird nun auf den Schwierigkeitsgrad 24 Karten anwenden. Zuerst wird die Sphärizität getestet. Das Ergebnis ist mit einem Signifikanzwert von 0,138 nicht signifikant, die Sphärizität ist also gegeben. Bei den Tests der Innersubjekteffekte bekommt man einen Signifikanzwert von 0,025. Dies bedeutet das es signifikante Unterschiede gibt. Man bildet nun die Differenzen zwischen den Interaktionen (eine Differenz pro Teilnehmer) und teilt diese durch die Anzahl der Teilnehmer. Bei diesen mittleren Differenzen gibt es 4 verschiedene signifikante Ergebnisse. Diese werden nun betrachtet.

Die mittlere Differenz *Annäherung* - Interaktionsart X ist immer negativ. Dies heißt dass *Annäherung* eine kleineren Mittelwert besitzt als alle anderen Interaktionsformen. Dies passt zu dem schon gemachten Aussage dass bei 24 Karten die Merkfähigkeit bei *Annäherung* am Besten ist. Die Interaktionsformen *Geräusche* und *Schütteln* haben die betragsmäßig größte signifikante mittlere Differenz zur erstplatzierten Interaktionsform *Annäherung*. Diese beiden wurden auch schon zuvor als schlechteste bei 24 Karten bewertet. Diese signifikanten Differenzen stärken diesen Befund noch weiter. Eine andere signifikante mittlere Differenz zu *Annäherung* hat *Telefon drehen*. Diese ist zwar mit -2,17 betragsmäßig nicht so gross wie bei *Geräusche* und *Schütteln* (-3,67), doch trotzdem erheblich.

Der vierte signifikante Wert bei den mittleren Differenzen ist zwischen den Inter-

aktionsformen *Keine Interaktion* und *Geräusche*. Der Unterschied von -2,83 bedeutet dass *Keine Interaktion* um einiges besser ist als *Geräusche*. *Keine Interaktion* wurde zuvor bei 24 Karten nach *Annäherung* als zweites bei der Merkfähigkeit platziert.

Bei den deskriptiven Werten gibt es erhebliche Unterschiede zwischen den Schwierigkeitsgraden. Bei zwölf Karten waren *Schütteln* und *Telefon drehen* am besten, und *Annäherung* am schlechtesten. Bei 24 Karten war *Annäherung* jedoch an der Spitze und *Geräusche* und *Schütteln* waren die letzten. Ein Grund für das schlechtere Abschneiden von *Schütteln* und *Telefon drehen* bei 24 Karten könnte die Ermüdung der Hand durch die dauernden Bewegungen sein. Die Teilnehmer sagten aber auch dass man sich die Karten wegen des Schüttelns des Gerätes schlechter merken kann. Ein Teilnehmer meinte die Struktur des Feldes, mit der er sich die Karten merke würde durch das Schütteln verloren gehen. Auch über das *Telefon drehen* sagte er dasselbe. Ein anderer Teilnehmer sagte dass es schwerer sei sich die Karten einzuprägen wenn man das Gerät umdrehte. Das gute Platzieren der *Annäherung* könnte an der Gewöhnung bei größeren Spielen liegen. Außerdem ist diese Interaktionsform nicht so anstrengend wie das Schütteln oder Umdrehen des Gerätes. Ein Teilnehmer sagte, das *Annäherung* leichter wird, wenn man genau ermittelt hat wo der Sensor am Gerät sitzt.

Doch es gibt noch einen anderen Faktor der sich auf die Merkfähigkeit auswirkt. Die Zeit die die Karten während einer Spielrunde durchschnittlich offen waren hat offensichtlich einen Einfluss. Wenn man eine Karte länger betrachtet kann man sie sich im Allgemeinen auch besser merken. Deshalb werden noch die Zeiten die die Karten durchschnittlich offen waren, betrachtet. Diese Werte sind in Abbildung 9 zu sehen.

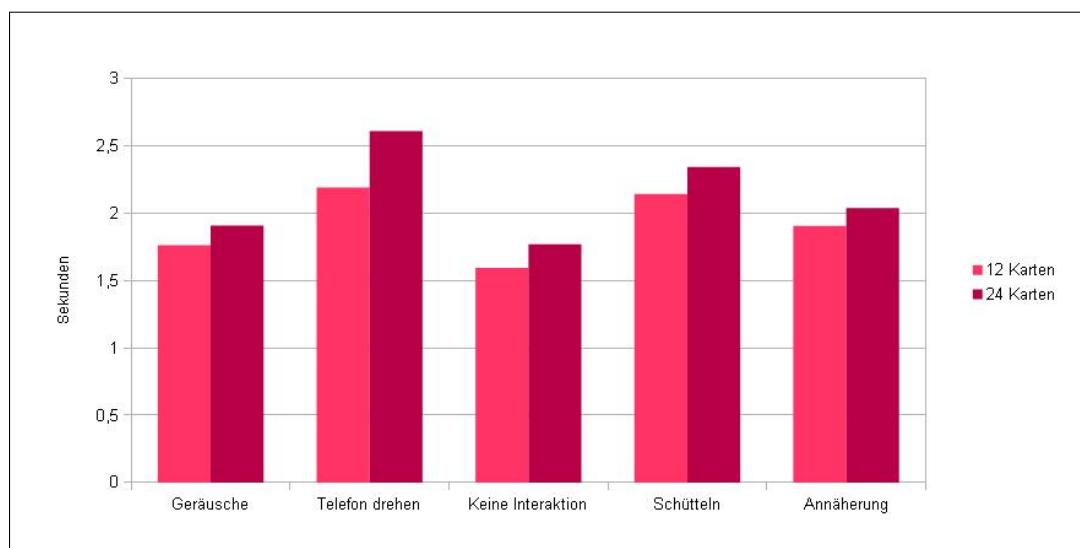


Abbildung 9: Zeit, die die Karten durchschnittlich geöffnet waren

Die Werte sind beim Schwierigkeitsgrad 24 Karten im Allgemeinen höher, *Annäherung* hat jedoch die kleinste Steigerung. Trotzdem hat *Annäherung* bei

24 Karten den ersten Platz bekommen. *Telefon drehen* hat den grössten Unterschied, was seine im Vergleich zu *Geräusche* besseren Platz bei 24 Karten etwas revidiert. *Keine Interaktion* hat niedrigere Werte als beispielsweise *Annäherung*, so dass *Keine Interaktion* näher an die Merkfähigkeit die bei *Annäherung* erreicht wurde, heranrückt. *Telefon drehen* und *Schütteln* haben die schlechtesten Werte, dies dämpft ihren Erfolg bei dem Schwierigkeitsgrad zwölf Karten und vergrössert ihren Misserfolg bei 24 Karten.

### 6.2 Ergebnisse des User Experience Fragebogens

Nachdem nun die Merkfähigkeit bei den verschiedenen Interaktionsformen untersucht wurde, soll noch die User Experience ermittelt werden. Denn was nützt eine Interaktionsform, mit der man sich Dinge leicht merken kann, wenn diese Interaktionsform eine schlechte User Experience erzielt?

Die Teilnehmer mussten zur jeder Interaktionsart einen UEQ Fragebogen ausfüllen. Der UEQ Fragebogen wurde von Laugwitz et al. entwickelt [29]. Er wurde entwickelt um die User Experience auf schnelle und einfache Weise zu messen. Die bipolaren Items wurden auf empirische Weise von Usability Experten ermittelt. Eine erste Version enthielt 80 Items, diese wurden später auf 26 reduziert. Die kurze Version des Fragebogens hat nur 20 Items. Jedes Item kann mit einem ganzzähligen Wert von eins (links) bis sieben (rechts) bewertet werden. Die Items sind aufgeteilt in die Gruppen Attraktivität, Stimulation, Originalität, Effizient und Durchschaubarkeit. Jeder Gruppe gehören dabei vier bipolare Paare an. Die vier Werte der Gruppen werden zusammengerechnet. Ein hoher Wert bei Originalität beispielsweise bedeutet dass der Befragte diese Interaktionsart für originell hielt. Jeder Teilnehmer musste während der Studie einen Fragebogen pro Interaktionsart ausfüllen. Bei 30 Teilnehmern hat man also 30 Fragebögen pro Interaktionsart. Die Gruppenergebnisse aller Fragebögen einer Interaktionsart werden summiert, Attraktivität mit Attraktivität, usw. Diese Gesamtwerte werden jeweils durch die Teilnehmerzahl und die Anzahl der Paare pro Gruppe dividiert. Die Ergebnisse sieht man in Abbildung 10.

Bei dieser Darstellung werden alle Paare einer Gruppe gleich gewichtet. Der Vorteil an dieser Darstellung ist, dass die Werte für jede Gruppe von eins bis sieben reichen, ganz wie im Fragebogen auch. Dies hilft dem Verständnis der Ergebnisse. Bei der Attraktivität ist die Interaktionsform *Keine Interaktion* vorne. Dies liegt wahrscheinlich an der Einfachheit und Bekanntheit der Interaktion. Eine Teilnehmerin meinte diese Interaktionsform hätte sich bewährt. Nur ein Teilnehmer hat sich von der Verzögerung, die beim Öffnen der Karten auftritt, genervt gefühlt. *Schütteln* und *Telefon drehen* gehören zu den weniger attraktiven Interaktionsformen. Dies liegt wohl daran, das man hier einiges an Kraft aufwenden muss. Mehrere Teilnehmer empfanden diese beiden Interaktionen als anstrengend und unangenehm und meinten es ginge ins Handgelenk. Nur ein Teilnehmer meinte dass *Schütteln* Spass mache und schnell sei. Die Interaktionsformen *Geräusche*

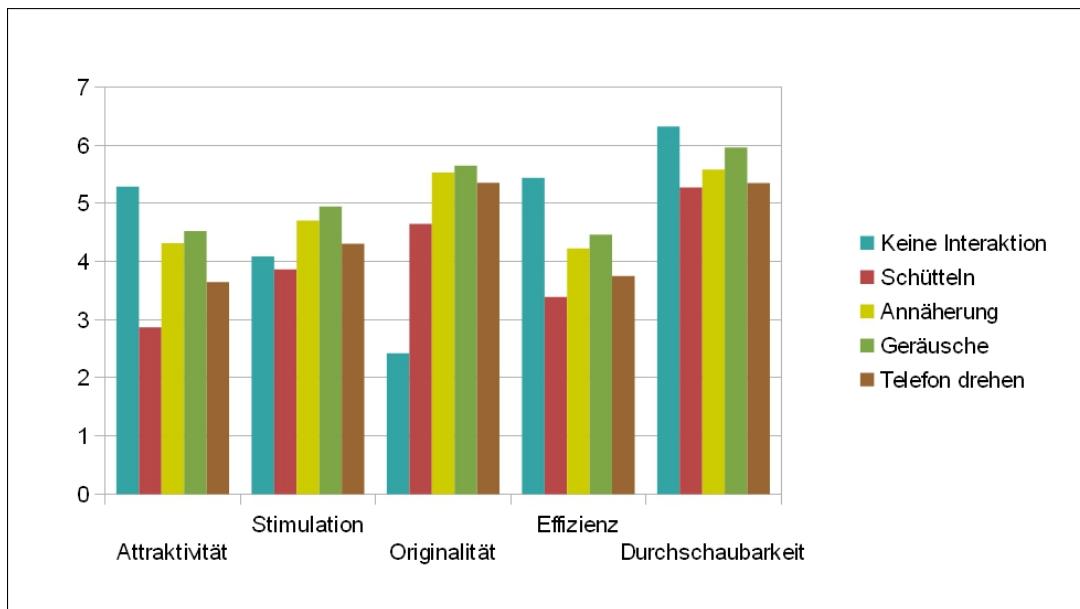


Abbildung 10: *Ergebnisse des UEQ Fragebogens als Durchschnittswerte*

und *Annäherung* wurden als ähnlich attraktiv bewertet. Ihre Werte sind nur etwas besser als der mittlere Wert (vier). Bei der Interaktionsart *Geräusche* hing es vom genutzten Geräusch ab, ob es als attraktiv empfunden wurde. Wurden die Karten durch Finger schnipsen geöffnet, wurde dies nach einer Weile anstrengend. Ein Teilnehmer meinte es sei einfacher mit der Zunge zu schnalzen als mit dem Finger zu schnipsen. Die besten Ergebnisse wurden erzielt wenn man kurze Worte zum Aufdecken der Karten gesagt hat. Diese Teilnehmer waren teilweise regelrecht begeistert von dieser Interaktionsform. Es wurde kommentiert dass Sprechen besser funktioniert als mit dem Finger zu schnipsen. Ein Teilnehmer hat trotz des Hinweises, dass kurze Geräusche sinnvoller sind, Dauergeräusche (ständiges auf den Tisch klopfen) gemacht. Die Karten waren dadurch nur kurz offen, doch der Teilnehmer war zufrieden. Auch wurde erwähnt, dass das Öffnen der Karten durch Geräusche Zuhause angenehm sei, nicht aber in der Öffentlichkeit. Die Interaktionsform *Annäherung* wurde im Allgemeinen als gut und interessant bezeichnet, nur ein Teilnehmer fand sie anstrengend.

Bei der Gruppe *Stimulation* sind die Werte der Interaktionsformen näher zusammen. *Geräusche* und *Annäherung* wurden als am Meisten stimulierend angesehen. Das Schütteln wurde nicht, wie man vielleicht glauben könnte, als stimulierend, sondern genau gegenteilig bewertet. Es landete somit bei der Stimulation auf dem letzten Platz. Auch *Keine Interaktion* schnitt nicht viel besser ab, und erreichte wie auch *Telefon drehen* einen Wert der etwas über der Mitte lag.

Bei der Gruppe *Originalität* schnitten die Interaktionsformen *Geräusche*, *Annäherung* und *Telefon drehen* mit Werten über 5 ziemlich gut ab. *Schütteln* lag mit etwa einem Punkt Abstand bei 4,65. Die Interaktionsform *Keine Interaktion* wurde nicht als besonders originell angesehen und bekam nur den Wert 2,43. Dies passt auch zu der schon erwähnten Aussage, dass *Keine Interaktion*, also nur Tippen auf den Bildschirm, sich schon bewährt hat.

Bei der Effizienz war *Keine Interaktion* wieder mit Abstand vorne. Es wurde

ein Wert von 5,44 erreicht. *Annäherung* und *Geräusche* kamen beide noch über vier, *Telefon drehen* und *Schütteln* waren aber darunter. Dies liegt wahrscheinlich daran, dass man bei *Schütteln* und *Telefon drehen* das Gerät bewegen muss. Wahrscheinlich wurden dadurch diese beiden Interaktionsformen als nicht besonders effizient empfunden.

Bei der Durchschaubarkeit lagen die Werte für alle Interaktionsformen über fünf. *Keine Interaktion* und *Geräusche* wurden als besonders Durchschaubar gehalten, die Werte lagen bei 6,33 und 5,97.

---

## 7 Zusammenfassung und Ausblick

Mit dieser Arbeit sollte die Frage des Einflusses verschiedener Interaktionsformen auf die Merkfähigkeit beantwortet werden. Die Interaktionsformen wurden mit einem Android Smartphone getestet. Dazu wurde ein Memor ähnliches Spiell entwickelt, welches mit Interaktion funktioniert. Die Auswahl der zu öffnenden Karten geschieht immer auf die gleiche Weise (durch antippen), das Öffnen passiert durch Interaktion. Es wurde 5 verschiedene Interaktionsformen getestet, diese sind: Schütteln des Gerätes, Umdrehen des Gerätes, Öffnen der Karten durch Geräusche, Abdeckung des Annäherungssensors und Keine Interaktion (also nur warten). Damit diese Interaktionsformen auf ihre Merkfähigkeit untersucht werden können, wurden Laborstudien durchgeführt. Es wurden 30 Personen zur Teilnahme eingeladen. Die Teilnehmer mussten das implementierte Spiel mit zwei verschiedenen Schwierigkeitsgraden, welche sich in der Anzahl der auf dem Feld befindlichen Karten unterschieden, spielen. Die Spielfelder hatten zwölf und 24 Karten. Die Spielergebnisse, wie beispielsweise, Anzahl der Versuche und die Zeit die jede Karte durchschnittlich offen war, wurde an einen Server geschickt. Außerdem mussten die Teilnehmer zu jeder Interaktionsform einen UEQ Fragebogen ausfüllen. Die Spieldaten und die UEQ Fragebögen wurden anschliessend ausgewertet.

Aus den Spielergebnissen wird geschlossen dass bei der Interaktionsform *Annäherung* die Merkfähigkeit am besten ist. Die Interaktionsform *Keine Interaktion* kommt an zweiter Stelle. Den letzten Platz belegt *Schütteln*. Diese Interaktionsart belegt bei dem einfachen Schwierigkeitsgrad zwar einen Spaltenplatz, doch bei vielen Karten wird ersichtlich, dass die Merkfähigkeit nicht vom Schütteln profitiert. Mehrere Teilnehmer sagten, dass man sich die Karten beim Schütteln schwerer merken könne.

Die Auswertung der UEQ Fragebögen ergab, dass *Keine Interaktion* bei den Gruppen Attraktivität, Effizienz und Durschaubarkeit die besten Werte hat. Die Stimulation liegt im Mittelfeld, während die Originalität sehr niedrig ist. *Geräusche* lag bei Stimulation und Originalität auf Platz eins und bei Attraktivität, Effizienz und Durchschaubarkeit auf Platz zwei. Die Interaktionsform *Schütteln* hat außer bei der Originalität immer den schlechtesten Wert.

Die Studie war mit einer Teilnehmerzahl von 30 eher klein. Außerdem wurde eine Laborstudie durchgeführt, so dass die externe Validität im Vergleich zu einer Feldstudie entsprechend geringer ist.

Als Lösung dieses Problems kann man den Google Play Market nutzen, um das Spiel einer riesigen Anzahl von potentiellen Teilnehmern zur Verfügung zu stellen. die Interaktionsform hatte vor allem bei 24 Karten eine sehr große Spannweite an Ergebnissen, es gab Geräusche die gut, und Geräusche die schlecht zum Aufdecken der Karten geeignet waren. Hatte ein Teilnehmer ein gut funktionierendes Geräusch gefunden, war er zufrieden mit der Interaktionsform. Ansonsten war es teilweise anstrengend wenn man ständig die Finger schnipsen musste. Man könnte in zukünftigen Untersuchungen verschiedene Geräusche vergleichen. Auch wäre eine Untersuchung mit vielen verschiedenen Geräten sinnvoll, um den Einfluss

der Qualität des Mikrofons zu ermitteln.

Ein weitere Interaktionsart die näher erforscht werden sollte, ist die *Annäherung*. Die Ergebnisse zeigen dass dies die Interaktionsform ist, die die Merkfähigkeit am besten unterstützt. Der Einsatz in Anwendungen, die das Merken von Informationen erfordern, sollte untersucht werden. Dabei könnte man auch den Effekt auf das freie Erinnern und auf das Wiedererkennen bestimmen.

Ein Teilnehmer gab an dass die Interaktionsform *Geräusche* nur zu Hause angenehm sei, nicht aber in der Öffentlichkeit. Man sollte also Interaktionsformen untersuchen, die ohne soziale Bedenken in der Öffentlichkeit ausgeführt werden können. Um Interaktionsformen für solch eine Untersuchungen zu finden, könnte man einen Studie durchführen, in der Teilnehmer die Aufgabe bekommen die Interaktionen in der Öffentlichkeit auszuführen, und gleichzeitig die Reaktionen der Mitmenschen zu beobachten. Die Teilnehmer könnten dann einen Fragebogen ausfüllen, um ihre Beobachtungen zu dokumentieren. Auch könnten kurze Fragen an die Mitmenschen gestellt werden um deren Einschätzung zu bekommen.

---

## 8 Anhang

Fragen zur Person:

Geschlecht:  weiblich  männlich

Alter: \_\_\_\_

Studiengang/Beruf: \_\_\_\_\_

Datum/Uhrzeit: \_\_\_\_\_

### **Einverständniserklärung**

Ich nehme an der Studie „**User experience for different interactions on a smartphone**“

freiwillig teil und bin darüber informiert worden, dass ich jederzeit die Teilnahme beenden kann.

Der Verwendung von Bildmaterial mit meiner Person in Präsentationen, Ausarbeitungen und Veröffentlichungen (auch online)

stimme ich zu  
 stimme ich nicht zu

Name: \_\_\_\_\_

Unterschrift: \_\_\_\_\_

**Bitte geben Sie Ihre Beurteilung ab.**

Bitte geben Sie nun Ihre Einschätzung ab. Kreuzen Sie bitte nur einen Kreis pro Zeile an.

	1	2	3	4	5	6	7		
unverständlich	<input type="radio"/>	verständlich	1						
kreativ	<input type="radio"/>	phantasielos	2						
leicht zu lernen	<input type="radio"/>	schwer zu lernen	3						
wertvoll	<input type="radio"/>	minderwertig	4						
langweilig	<input type="radio"/>	spannend	5						
uninteressant	<input type="radio"/>	interessant	6						
schnell	<input type="radio"/>	langsam	7						
originell	<input type="radio"/>	konventionell	8						
gut	<input type="radio"/>	schlecht	9						
kompliziert	<input type="radio"/>	einfach	10						
herkömmlich	<input type="radio"/>	neuartig	11						
unangenehm	<input type="radio"/>	angenehm	12						
aktivierend	<input type="radio"/>	einschläfernd	13						
ineffizient	<input type="radio"/>	effizient	14						
übersichtlich	<input type="radio"/>	verwirrend	15						
unpragmatisch	<input type="radio"/>	pragmatisch	16						
aufgeräumt	<input type="radio"/>	überladen	17						
attraktiv	<input type="radio"/>	unattraktiv	18						
unsympathisch	<input type="radio"/>	sympathisch	19						
konservativ	<input type="radio"/>	innovativ	20						

**Genutzte Interaktion**

Keine	Schütteln	Annäherung	Geräusche	Telefon drehen



## Literatur

- [1] Pär-Anders Albinsson and Shumin Zhai. High precision touch screen interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 105–112, New York, NY, USA, 2003. ACM.
- [2] Florian Alt, Stefan Schneegass, Michael Girgis, and Albrecht Schmidt. Cognitive effects of interactive public display applications. In *Proceedings of the 2nd ACM International Symposium on Pervasive Displays*, PerDis '13, pages 13–18, New York, NY, USA, 2013. ACM.
- [3] Florian Alt, Stefan Schneegaß, Albrecht Schmidt, Jörg Müller, and Nemanja Memarovic. How to evaluate public displays. In *Proceedings of the 2012 International Symposium on Pervasive Displays*, PerDis '12, pages 17:1–17:6, New York, NY, USA, 2012. ACM.
- [4] J. R. Anderson. *Kognitive Psychologie (3. Auflage)*. Spektrum, Heidelberg, 2001.
- [5] R. C. Atkinson and R. M. Shiffrin. Human memory: A proposed system and its control processes. In K. W. Spence and J. T. Spence (Eds.), *The Psychology of learning and motivation: Advances in research and theory (vol. 2)*, pages 89 – 105, 1968.
- [6] Hanif Baharin and Ralf Mühlberger. Living with the sound of the past: experiencing sonic atomic interaction using the sound diary. In *Proceedings of the 10th International Conference NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction*, CHINZ '09, pages 101–104, New York, NY, USA, 2009. ACM.
- [7] Matthias Baldauf, Sebastian Zambanini, Peter Fröhlich, and Peter Reichl. Markerless visual fingertip detection for natural mobile device interaction. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 539–544, New York, NY, USA, 2011. ACM.
- [8] Shruti Bhandari and Youn-Kyung Lim. Exploring gestural mode of interaction with mobile phones. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pages 2979–2984, New York, NY, USA, 2008. ACM.
- [9] Stephen A. Brewster and Michael Hughes. Pressure-based text entry for mobile devices. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, pages 9:1–9:4, New York, NY, USA, 2009. ACM.
- [10] Christina Dicke, Katrin Wolf, and Yaroslav Tal. Foogue: eyes-free interaction for smartphones. In *Proceedings of the 12th international conference on*

*Human computer interaction with mobile devices and services*, MobileHCI '10, pages 455–458, New York, NY, USA, 2010. ACM.

- [11] Thomas J. Donahue, G. Michael Poor, Martez E. Mott, Laura Marie Leventhal, Guy Zimmerman, and Dale Klopfer. On interface closeness and problem solving. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, TEI '13, pages 139–146, New York, NY, USA, 2013. ACM.
- [12] Michael Feld, Saeedeh Momtazi, Farina Freigang, Dietrich Klakow, and Christian Müller. Mobile texting: can post-asr correction solve the issues? an experimental study on gain vs. costs. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, IUI '12, pages 37–40, New York, NY, USA, 2012. ACM.
- [13] Leah Findlater, Jon E. Froehlich, Kays Fattal, Jacob O. Wobbrock, and Tanya Dastyar. Age-related differences in performance with touchscreens compared to traditional mouse input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 343–346, New York, NY, USA, 2013. ACM.
- [14] Vittorio Fuccella, Poika Isokoski, and Benoit Martin. Gestures and widgets: performance in text editing on multi-touch capable mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2785–2794, New York, NY, USA, 2013. ACM.
- [15] D. R. Godden and A. D. Baddeley. Context-dependent memory in two natural environments: On land and under water. *British Journal of Psychology*, 66:325–331, 1975.
- [16] Teng Han, Jason Alexander, Abhijit Karnik, Pourang Irani, and Sriram Subramanian. Kick: investigating the use of kick gestures for mobile interactions. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 29–32, New York, NY, USA, 2011. ACM.
- [17] Seongkook Heo and Geehyuk Lee. Forcedrag: using pressure as a touch input modifier. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*, OzCHI '12, pages 204–207, New York, NY, USA, 2012. ACM.
- [18] Eve Hoggan, Stephen A. Brewster, and Jody Johnston. Investigating the effectiveness of tactile feedback for mobile touchscreens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1573–1582, New York, NY, USA, 2008. ACM.
- [19] Sungjae Hwang, Myungwook Ahn, and Kwangyun Wohn. Magnetic marionette: magnetically driven elastic controller on mobile device. In *Proceedings*

*of the companion publication of the 2013 international conference on Intelligent user interfaces companion*, IUI '13 Companion, pages 75–76, New York, NY, USA, 2013. ACM.

- [20] Clare-Marie Karat, Christine Halverson, Daniel Horn, and John Karat. Patterns of entry and correction in large vocabulary continuous speech recognition systems. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, CHI '99, pages 568–575, New York, NY, USA, 1999. ACM.
- [21] Spencer D. Kelly, Dale J. Barr, R. Breckinridge Church, and Katheryn Lynch. Offering a hand to pragmatic understanding: The role of speech and gesture in comprehension and memory. *Journal of Memory and Language*, 40(4):577 – 592, 1999.
- [22] Hamed Ketabdar, Hengwei Chang, Peyman Moghadam, Mehran Roshandel, and Babak Naderi. Magiguitar: a guitar that is played in air! In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services companion*, MobileHCI '12, pages 181–184, New York, NY, USA, 2012. ACM.
- [23] Hamed Ketabdar, Peyman Moghadam, Babak Naderi, and Mehran Roshandel. Magnetic signatures in air for mobile devices. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services companion*, MobileHCI '12, pages 185–188, New York, NY, USA, 2012. ACM.
- [24] Hamed Ketabdar, Mehran Roshandel, and Kamer Ali Yüksel. Magiwrite: towards touchless digit entry using 3d space around mobile devices. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, MobileHCI '10, pages 443–446, New York, NY, USA, 2010. ACM.
- [25] Hamed Ketabdar, Kamer Ali Yüksel, and Mehran Roshandel. Magitact: interaction with mobile devices based on compass (magnetic) sensor. In *Proceedings of the 15th international conference on Intelligent user interfaces*, IUI '10, pages 413–414, New York, NY, USA, 2010. ACM.
- [26] Kenrick Kin, Maneesh Agrawala, and Tony DeRose. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *Proceedings of Graphics Interface 2009*, GI '09, pages 119–124, Toronto, Ont., Canada, Canada, 2009. Canadian Information Processing Society.
- [27] Sven Kratz and Michael Rohs. Hoverflow: expanding the design space of around-device interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, pages 4:1–4:8, New York, NY, USA, 2009. ACM.

- [28] Jason Kurczak, T. C. Nicholas Graham, Claire Joly, and Regan L. Mandryk. Hearing is believing: evaluating ambient audio for location-based games. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, ACE '11, pages 32:1–32:8, New York, NY, USA, 2011. ACM.
- [29] Bettina Laugwitz, Theo Held, and Martin Schrepp. Construction and evaluation of a user experience questionnaire. In *Proceedings of the 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society on HCI and Usability for Education and Work*, USAB '08, pages 63–76, Berlin, Heidelberg, 2008. Springer-Verlag.
- [30] Hao Lü and Yang Li. Gesture avatar: a technique for operating mobile user interfaces using gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 207–216, New York, NY, USA, 2011. ACM.
- [31] George Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information, 1956. One of the 100 most influential papers in cognitive science: <http://cogsci.umn.edu/millennium/final.html>.
- [32] Matei Negulescu, Jaime Ruiz, Yang Li, and Edward Lank. Tap, swipe, or move: attentional demands for distracted smartphone input. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 173–180, New York, NY, USA, 2012. ACM.
- [33] Hugo Nicolau and Joaquim Jorge. Elderly text-entry performance on touchscreens. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '12, pages 127–134, New York, NY, USA, 2012. ACM.
- [34] Jaime Ruiz, Yang Li, and Edward Lank. User-defined motion gestures for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 197–206, New York, NY, USA, 2011. ACM.
- [35] Frank P. Schulte, Edgar Heineken, and Simon Gall. What's in it for product marketing when using a pageflip in online-catalogues? *i-com*, pages 40–46, Feb, 2009.
- [36] Sajad Shirali-Shahreza, Gerald Penn, Ravin Balakrishnan, and Yashar Ganjali. Seesay and hearsay captcha for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2147–2156, New York, NY, USA, 2013. ACM.
- [37] Steven M. Smith. Remembering in and out of context. *Journal of Experimental Psychology: Human Learning and Memory*, 5(5):460–471, Sep, 1979.

- [38] Vytautas Vaitukaitis and Andreas Bulling. Eye gesture recognition on portable devices. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 711–714, New York, NY, USA, 2012. ACM.
- [39] Keith Vertanen and Per Ola Kristensson. Parakeet: a continuous speech recognition system for mobile touch-screen devices. In *Proceedings of the 14th international conference on Intelligent user interfaces*, IUI '09, pages 237–246, New York, NY, USA, 2009. ACM.
- [40] Daniel Vogel and Patrick Baudisch. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 657–666, New York, NY, USA, 2007. ACM.
- [41] Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. Lucid touch: a see-through mobile device. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pages 269–278, New York, NY, USA, 2007. ACM.
- [42] Katrin Wolf, Christina Dicke, and Raphael Grasset. Touching the void: gestures for auditory interfaces. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, TEI '11, pages 305–308, New York, NY, USA, 2011. ACM.



## Abbildungsverzeichnis

1	Hier sieht man die Activities der App und deren Verbindungen. Grüne Verbindungen werden automatisch und als erstes verfolgt. . . . .	34
2	Hier ist eine laufende Spielrunde zu sehen. Es sind zwei Karten geöffnet, und mehrere Paare wurden schon gefunden (freie Stellen). In der Titelleiste sieht man Informationen wie beispielsweise die Spielzeit. . . . .	39
3	Spieldatenbildschirm (CardsAxB) und davon genutzte Komponenten. Die Activity ist in gelb gehalten, die genutzten Threads werden in rot dargestellt, Listener sind grün eingefärbt und Handler haben eine hellblaue Farbe . . . . .	45
4	Die HighscoreActivity mit Einträgen. Es sind fast nur die Listen für 3x4 und 4x6 gefüllt, da diese Spieldatengrößen in der Studie verwendet wurden. . . . .	46
5	Hier können Einstellungen zur App gemacht werden. . . . .	47
6	Ein Teilnehmer beim Interagieren mit dem Gerät. Die Interaktionsform ist <i>Keine Interaktion</i> . . . . .	52
7	<i>Boxplot mit zusätzlicher Angabe von Mittelwert und Standardabweichung für Anzahl der Versuche - zwölf Karten</i> . . . . .	55
8	<i>Boxplot mit zusätzlicher Angabe von Mittelwert und Standardabweichung für Anzahl der Versuche - 24 Karten</i> . . . . .	57
9	<i>Zeit, die die Karten durchschnittlich geöffnet waren</i> . . . . .	59
10	<i>Ergebnisse des UEQ Fragebogens als Durchschnittswerte</i> . . . . .	61

## ABBILDUNGSVERZEICHNIS

## Listings

1	Hinzufügen eines TimerThreads in die timerlist mit nachfolgendem Starten des Timerthreads . . . . .	43
2	Test ob der aktuelle TimerThread der einzige ist. Wenn ja wird Nachricht an den OpenCardsHandler geschickt . . . . .	44

## LISTINGS

---

## 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 24. Juli 2013 \_\_\_\_\_