

Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit Nr. 133

**Nutzerzentrierte Datenerhebung
zur Aktivitätserkennung von
alltäglichen Tätigkeiten durch
mobile Geräte**

Halil Bugdayci

Studiengang: Softwaretechnik

Prüfer/in: Jun.-Prof. Dr. Niels Henze

Betreuer/in: Stefan Schneegaß M.Sc.

Beginn am: 12. Mai 2014

Beendet am: 11. November 2014

CR-Nummer: H.5.2

Kurzfassung

Die Aktivitätserkennung mit einem Smartphone ist durch die Sensoren, die im Smartphone integriert sind, möglich. Dazu werden keine externen Sensoren benötigt. Um eine Aktivität zu erkennen, müssen die Daten, die von den Sensoren geliefert werden, ausgewertet werden. Dafür werden Trainingsdaten, die während einer Aktivität aufgezeichnet werden, benötigt, die das Erkennen der Aktivität erst möglich machen.

Diese Arbeit beschäftigt sich mit der nutzerzentrierten Datenerhebung zur Aktivitätserkennung. Die benötigten Sensordaten werden mit Hilfe einer mobilen Android Anwendung von Personen gesammelt. Die mobile Android Anwendung wurde hierfür in Form eines Spiels entwickelt und verbreitet, sodass es von vielen Personen benutzt wird. Mit dieser Forschungsmethode soll erreicht werden, dass die Sensordaten für die Aktivitätserkennung von vielen Personen erhoben werden können.

Abstract

Activity recognition with a smartphone is possible with the integrated sensors. So there is no need of external sensors. To recognize an activity, the data obtained from the sensors have to be analyzed. Therefore, training data which is recorded during an activity is needed.

This work deals with user-centered data collection for activity recognition on mobile devices. The required sensor data are collected from people by using a mobile Android application. The mobile Android application is developed in the form of a game so that many persons use it. The aim is to collect many sensor data for activity recognition.

Inhaltsverzeichnis

1	Einleitung	7
1.1	Motivation	8
1.2	Aufbau des Dokuments	8
2	Grundlagen	9
2.1	Aktivitätserkennung	9
2.2	Research in the Large	16
3	Das Konzept	19
4	Acremode	23
4.1	Beschreibung	23
4.2	Architektur	26
4.3	Spezifikation	26
5	Studie über Distanzmessung und Nutzerbewertungen	39
5.1	Verlässlichkeit der Distanzmessung	39
5.2	Nutzerbewertung	40
6	Auswertung der Sensordaten	43
6.1	Ausrichtung des Smartphones	43
6.2	Gesammelte Sensordaten	44
7	Zusammenfassung und Ausblick	45
	Literaturverzeichnis	47

1 Einleitung

Die Relevanz der Aktivitätserkennung durch mobile Endgeräte ist in vielen unterschiedlichen Einsatzgebieten sehr hoch. Ein Einsatzgebiet der Aktivitätserkennung ist im Bereich der Gesundheit. In diesem Gebiet werden regelmäßig generierte Daten von Patienten durch Smartphones gesammelt. Diese Methode hilft den Ärzten Krankheiten oder bestimmte Symptome zu beobachten und zu diagnostizieren [1]. Vom sozialen Netzwerk bis zur Erkennung von Unfällen ist die Aktivitätserkennung durch mobile Smartphones relevant, da mittlerweile das Benutzen eines Smartphones zu den alltäglichen Tätigkeiten wie das Gehen, Laufen, Sitzen und Stehen gehört [2, 3]. Dabei handelt es sich auch um das Surfen im Netz, Chatten mit Freunden, Lesen von E-Mails und das soziale Netzwerk.

Die oben genannten alltäglichen Aktivitäten können durch Sensoren, die im Smartphone integriert sind, erkannt werden. Sensoren sind technische Bauteile, die bestimmte Eigenschaften qualitativ messen können. Je nach Art des Sensors kann es sich um Eigenschaften wie der Druck, die Helligkeit, die Temperatur oder die Beschleunigung handeln. Ein Fall-Erkennungs- und Benachrichtigungssystem kann im Gesundheitswesen verwendet werden, indem Sensoren eingesetzt werden, welches das Fallen einer Person mit Gesundheitsproblemen feststellt und diesen Fall an die zugeordnete Stelle mitteilt [4]. Dazu wird der im Smartphone integrierte Beschleunigungssensor und Gyroskop verwendet. Da diese Sensoren in vielen Smartphones integriert sind, werden keine externen Sensoren benötigt, die an die Personen befestigt werden müssen, um bestimmte Aktivitäten, wie beispielsweise in der oben genannten Untersuchung den Fall einer Person, zu erkennen. Smartphones besitzen neben den Bewegungssensoren, wie dem Beschleunigungssensor und dem Gyroskop, weitere Sensoren, wie Umgebungssensoren und Positionssensoren.

Die Datenerhebung von bestimmten Daten durch mobile Endgeräte spielt eine immer wichtigere Rolle in Forschungen und Studien über Mensch-Computer-Interaktion. Diese Studien werden häufig in einer kleinen und kontrollierten Umgebung durchgeführt. Die Ergebnisse können deshalb in Bezug auf das Verhalten von Nutzern im realen Zusammenhang nicht verallgemeinert werden und somit mangelt es an externer Aussagekraft [5]. Um dieses Problem anzugehen kann ein App Store, eine Vertriebsplattform für App's, verwendet werden, womit auch die interne Aussagekraft mit der externen Aussagekraft ausgetauscht wird [6]. Dadurch wird eine potenziell sehr hohe Anzahl an Nutzer erreicht. In diesem Fall sind diese Nutzer die Teilnehmer der Studie.

Ein Beispiel für eine derartige Studie ist die Studie über eine neu entwickelte Interaktionstechnik. Wenn eine hohe Anzahl an Teilnehmern erreicht wird, kann ein Feedback über die Interaktionstechnik erhalten werden und es wird somit eine starke Aussagekraft über die Studie erreicht [6].

1.1 Motivation

Es gibt bereits Anwendungen und Sensoren zur Aktivitätserkennung, die in den meisten Smartphones integriert sind. Dazu gehören zum Beispiel der Schrittzähler und das GPS. Es bietet sich die Möglichkeit, diese Anwendungen oder Sensoren bei Studien zu verwenden. Es ist sehr mühsam, nutzerzentrierte Daten zu sammeln [7]. Das würde das aufwendige Sammeln der Daten erleichtern. Je größer die Anzahl der Teilnehmer und somit die Anzahl der gesammelten Daten, desto repräsentativer ist das Ergebnis nach der Auswertung. Da die Anzahl der Teilnehmer und der gesammelten Daten ein wichtiger Faktor für die Studie ist, muss der Forscher sich eine Methode überlegen, um Daten von möglichst vielen Teilnehmern zu sammeln.

Die Anzahl der Smartphone-Nutzer nimmt kontinuierlich zu. Es ist vorteilhaft für den Forscher, der die Studie durchführt, die Daten über die Smartphones der Nutzer zu sammeln, weil mit weniger Aufwand mehr Menschen erreicht werden und an der Studie teilnehmen. Die Datensammlung über die Smartphones erfolgt durch Apps, die nach den Bedürfnissen der Studie und Forscher entwickelt werden. Das Problem bei der Studie ist, dass viele Teilnehmer benötigt werden, die die App installieren.

Im Rahmen der Bachelorarbeit wird ein Spiel für die Datenerhebung zur Aktivitätserkennung durch Smartphones entwickelt. Die Datenerhebung folgt durch die Installation und Benutzung der App durch den Nutzer. Weil die App in Form eines Spiels entwickelt wird, steigt die Interesse und der Mehrwert für den Nutzer. Der Nutzer kann spielend bestimmte Tätigkeiten ausführen, wobei diese mit der geeigneten Metrik gemessen und anschließend gespeichert bzw. an den Forscher weitergeleitet werden. Die Aufgabe besteht darin, die App zu entwickeln und zu verbreiten. Durch diesen Ansatz sollen leicht neue Spiele entwickelt werden, um neue Fragestellungen zu lösen.

1.2 Aufbau des Dokuments

In Kapitel 2 wird auf die Grundlagen sowie die verwandten Arbeiten eingegangen. Kapitel 2.1 befasst sich mit den für die Aktivitätserkennung relevanten Grundlagen. Die Sensoren des mobilen Smartphones werden näher beschrieben. In Kapitel 2.2 wird auf die benutzte Methodik für die Studie eingegangen. Kapitel 3 befasst sich mit der konzeptuellen Umsetzung der Datenerhebung zur Aktivitätserkennung. Die App wird in Kapitel 4 spezifiziert und näher beschrieben. Kapitel 5 beschreibt die mit der App durchgeführte Studie und Kapitel 6 behandelt die Auswertung.

2 Grundlagen

In diesem Abschnitt werden zuerst Anwendungsbeispiele bezüglich der Aktivitätserkennung und die Aktivitätserkennung beschrieben. Dazu gehört auch die Sensorausstattung, die die Aktivitätserkennung erst möglich macht. Anschließend wird auf das Thema *Research in the Large* in Kapitel 2.2 eingegangen.

2.1 Aktivitätserkennung

Die Aktivitätserkennung identifiziert die Aktivität, die eine Person ausführt [1]. Es gibt viele Untersuchungen über die Aktivitätserkennung, die auf der Sensorausstattung von mobilen Smartphones basieren. Eine andere Möglichkeit, solche Untersuchungen durchzuführen, ist das Anbringen von mehreren Sensoren an verschiedenen Körperteilen der beteiligten Personen. Die erhobenen Daten werden an einen nahegelegenen Server übermittelt [8].

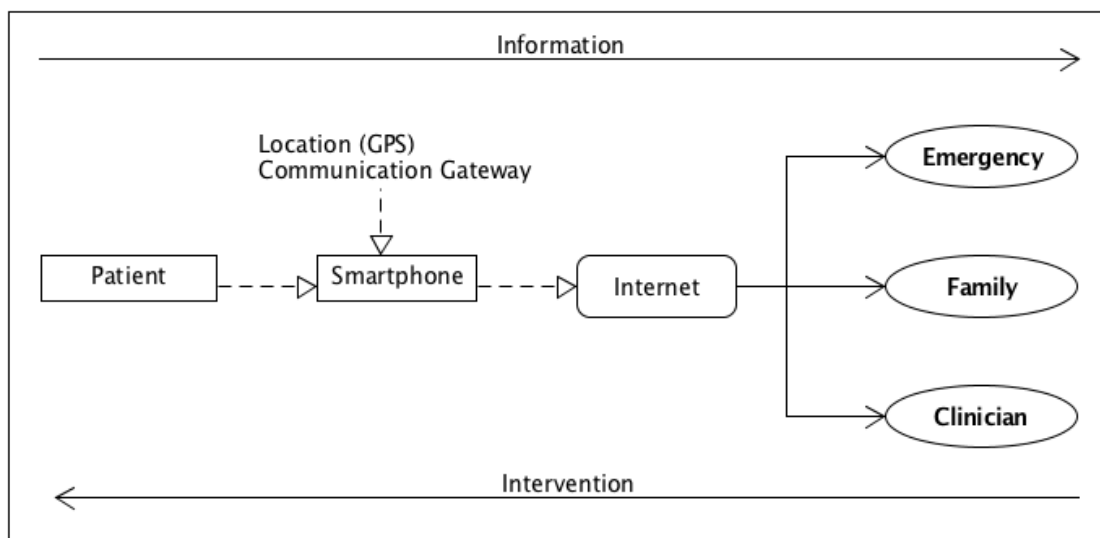


Abbildung 2.1: Überwachung der Gesundheit von älteren Patienten [8]

In Abbildung 2.1 ist der Ablauf beschrieben, wie Patienten mithilfe des Smartphones beobachtet werden und wie in Notfällen eingegriffen wird. Es werden an einem Patienten verschiedene Sensoren befestigt, die die Bewegungen vom Patienten aufzeichnen. In diesem Fall sind es zwei Bewegungssensoren und ein Sensor für die Herz- und Atemfrequenz. Das Smartphone ermittelt die genaue Position

des Patienten. Die aufgezeichneten Daten werden über das Internet je nach Situation an die betroffene Stelle weitergeleitet. Dies kann der Notruf, der Arzt oder die Familie sein.

W. Wibisono et al. haben ein Fall-Erkennungs-System beschrieben [4]. Das Fall-Erkennungs-Algorithmus verwendet die Sensorwerte vom Beschleunigungssensor des Smartphones. Mit Hilfe dieses Systems kann der Fall einer Person erkannt werden. Dies könnte beispielsweise in einer ähnlich wie in Abbildung 2.1 beschriebenen Umgebung verwendet werden, ohne externe Sensoren an den Patienten befestigen zu müssen. Denn W. Wibisono et al. ersetzen die in Abbildung 2.1 verwendeten externen Bewegungssensoren durch die Sensoren des Smartphones.

Mobile Endgeräte wie Smartphones enthalten unterschiedliche Sensoren. GPS Sensoren, Sprachsensoren wie z.B. das Mikrofon, Bildsensoren, Lichtsensoren, Temperatursensoren, der Kompass und Beschleunigungssensoren gehören zur Sensorausstattung eines Smartphones. Smartphones sind praktisch und besitzen eine starke Rechenleistung. Sie können Daten senden und empfangen. Diese Eigenschaften ermöglichen es, Smartphones als Datengewinnungsquelle in Forschungen zu verwenden. [9]

J.R. Kwapisz et al. haben Daten von einer Testgruppe gesammelt. Die Personen haben vorgeschriebene Aktivitäten durchgeführt, sodass nachträglich die gesammelten Daten für die Erkennung der vorgeschriebenen Aktivität verwendet wurden. Diese Daten dienen letztendlich zum Erlernen der Erkennung der Aktivität. Zu den Standard-Aktivitäten gehören beispielsweise das Gehen, das Treppensteigen, das Sitzen und das Stehen. Es wurde eine App für das Verwalten der Daten entwickelt, die eine grafische Oberfläche hat und die Eingabe für die Benennung der Aktivität sowie den Namen des Teilnehmers akzeptiert.

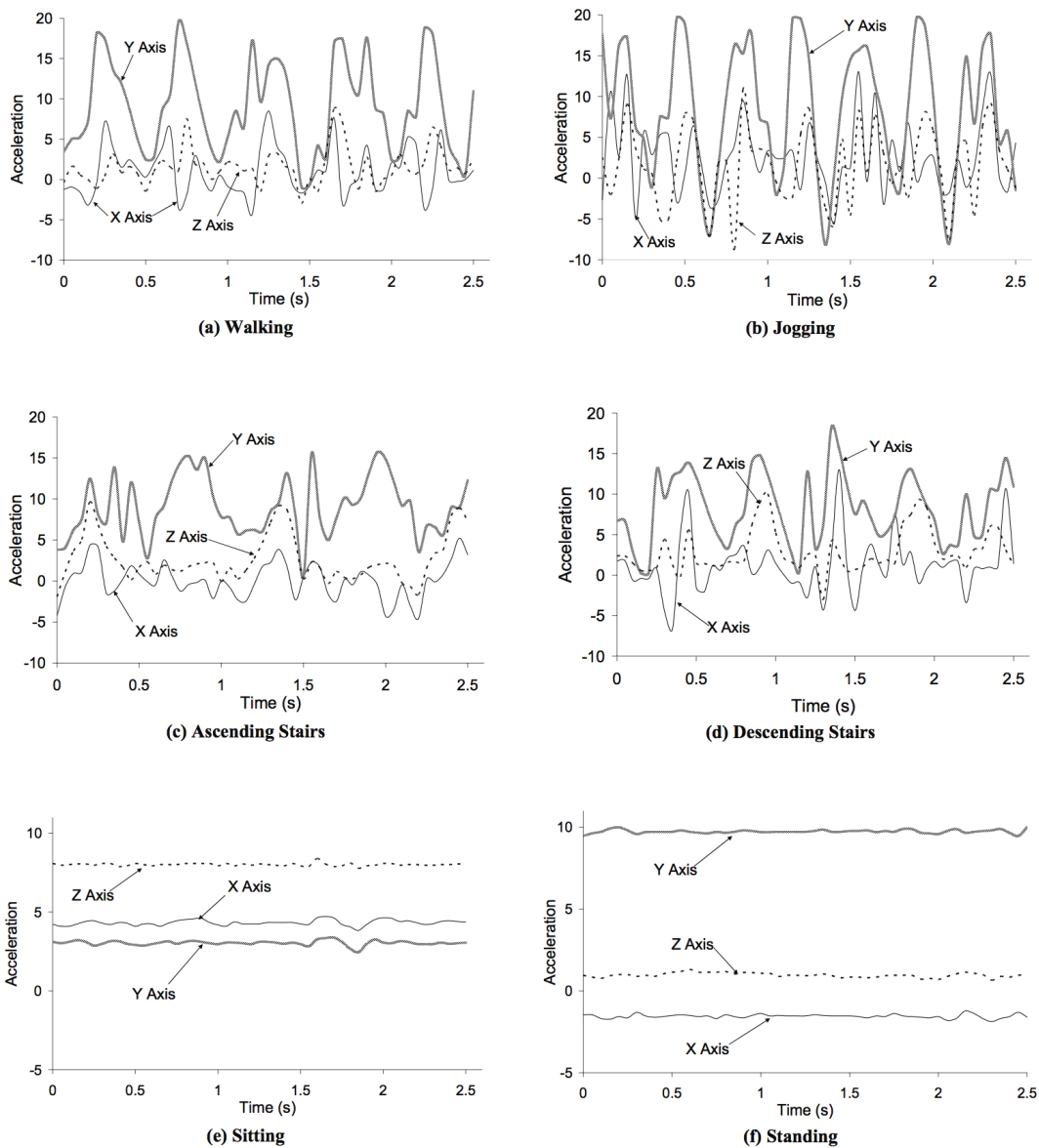


Abbildung 2.2: Grafische Darstellung der Beschleunigung für das Gehen, Joggen, Treppensteigen, Sitzen und Stehen [9]

In Abbildung 2.2 ist die grafische Darstellung der Daten für die oben genannten Aktivitäten, die J.R.Kwapisz et al. gesammelt und erkannt haben.

Um eine Aktivität zu erkennen werden die relevanten Daten betrachtet oder in bestimmte Algorithmen eingesetzt. In den meisten Fällen müssen die Roh-Daten gefiltert und erst danach in den Algorithmus

zur Aktivitätserkennung eingesetzt werden. Die Daten werden von Sensoren aufgezeichnet. Die Sensoren sind also die Grundlage für die Erkennung von Aktivitäten. Aus den Sensoren resultierende Informationen machen eine Aktivitätserkennung erst möglich.

Im folgenden Kapitel werden diese Sensoren in Android-Smartphones einzeln erläutert.

Sensorausstattung

Das Android Betriebssystem bietet verschiedene Sensoren an. Diese sind in der Lage Roh-Daten sehr präzise zu messen und liefern somit Informationen über die Position des Smartphones und darüber, wie das Smartphone bewegt wird. Android bietet eine Schnittstelle, um die Sensoren zu verwenden und an Informationen zu gelangen. Es gibt zwei unterschiedliche Sensorarten, die in Android angeboten werden. Dabei handelt es sich einmal um Sensoren, die auf einer Hardware basieren. Diese sind technische Komponenten, also Hardware, die im Smartphone eingebaut sind. Die Messung wird in diesem Fall von diesem Sensor durchgeführt. Es gibt auch software-basierte Sensoren. Hier handelt es sich nicht um direkte Hardware. Die Messung wird beim software-basierten Sensor von anderen, hardware-basierten Sensoren abgeleitet und durchgeführt.

In Tabelle 2.1 ist die Übersicht zu den in Android verfügbaren Sensoren aufgelistet. In der ersten Spalte steht der Sensor und in der zweiten Spalte der Sensortyp, das heißt, ob es hardware- oder software-basiert ist. Die dritte Spalte beinhaltet Informationen über die Messung des Sensors wie beispielsweise die Maßeinheit.

Es sind 13 Sensoren aufgelistet. Jedoch sind nur einige davon für die Aktivitätserkennung relevant. Diese werden in diesem Kapitel bezüglich folgender Aspekte betrachtet:

- Funktion
- Einschränkungen
- Vor- und Nachteile

Beschleunigungssensor

Der Beschleunigungssensor misst die Beschleunigung, die das Gerät erfährt. Aus dem Koordinatensystem können drei Werte aus den drei physikalischen Achsen, X, Y und Z ausgelesen werden. Die Erdanziehungskraft ist in diesen Werten enthalten. Als Einheit wird $\frac{m}{s^2}$ benutzt.

Die Werte werden in einer vom Entwickler konfigurierten Abtastgeschwindigkeit vom Beschleunigungssensor ausgelesen. Es gibt vier vordefinierte Konstanten in Android. `SENSOR_DELAY_NORMAL` ist die Standard-Abtastgeschwindigkeit und beträgt 200 Millisekunden. Die etwas schnellere Abtastgeschwindigkeit definiert die Konstante `SENSOR_DELAY_UI` mit 60 Millisekunden. Die Konstante `SENSOR_DELAY_GAME` hat den Wert 20 Millisekunden und die Konstante `SENSOR_DELAY_FASTEST` hat den Wert 0 Millisekunden. Je nach Anwendungszweck wird die Abtastgeschwindigkeit passend ausgewählt. Ab Android 3.0 (API Level 11) gibt es auch die Möglichkeit, die Abtastgeschwindigkeit

manuell als einen absoluten Wert in Mikrosekunden anzugeben. Diese Angabe wird als eine vorgeschlagene Abtastgeschwindigkeit angenommen. Denn das Betriebssystem Android oder möglicherweise andere Anwendungen, die unter demselben Betriebssystem laufen, können die Abtastgeschwindigkeit abändern. Android benutzt üblicherweise eine schnellere Abtastgeschwindigkeit als die angegebene Geschwindigkeit. Daher sollte je nach Bedarf die möglichst langsame Abtastgeschwindigkeit angegeben werden. Falls aus beliebigen Gründen die Abtastgeschwindigkeit geändert werden soll, muss der im *Sensor Manager* registrierte *Sensor Event Listener* abgemeldet werden und erneut mit der geänderten Abtastgeschwindigkeit registriert werden. Je langsamer die Abtastgeschwindigkeit, desto weniger wird der Prozessor belastet. Somit verbraucht das Smartphone weniger Energie. Wenn keine Sensorwerte mehr benötigt werden, sollte der *Sensor Event Listener* vom *Sensor Manager* abgemeldet werden, um Energie zu sparen. Sonst würde es bei Sensoren mit hohem Leistungsbedarf den Akku des Smartphones schneller leeren.

Um Werte vom Beschleunigungssensor zu erhalten, muss im *Sensor Manager* ein *Sensor Event Listener* für die *Sensor Events* des Beschleunigungssensors registriert werden. Das heißt kurzgefasst, dass die Werte vom Beschleunigungssensor, die sich ändern, erhalten werden. Über den *Sensor Manager* wird der Zugriff auf die Sensoren des Smartphones ermöglicht. Der *Sensor Event Listener* wird selbst definiert. Es gibt Methoden, die überschrieben werden müssen. Diese Methoden werden ausgelöst, wenn zum Beispiel der Wert des Beschleunigungssensors sich verändert. In diesem Fall wird die Methode *onSensorChanged(SensorEvent event)* ausgelöst. Dabei wird das entsprechende *Sensor Event* als Parameter durchgegeben. Aus diesem *Sensor Event* vom Beschleunigungssensor können folgende Werte ausgelesen werden:

Beschleunigung auf der X, Y und Z-Achse inklusive der Erdanziehungskraft, der Zeitstempel, die Genauigkeit des *Sensor Events* und der Sensor, von dem das *Sensor Event* geschickt wurde. Hier ist es der Beschleunigungssensor.

Falls die Erdanziehungskraft außer Acht gelassen werden soll, muss es manuell aus den gegebenen Werten herausgefiltert werden. Die Zeitstempel können dafür verwendet werden, um die Geschwindigkeit, wie oft der Sensor *Sensor Events* schickt, zu messen. Denn es wird von Android keine öffentliche Methode zur Verfügung gestellt, die die Geschwindigkeit ausgibt bzw. berechnet. Das könnte als ein Nachteil gesehen werden. Es kann nicht damit gerechnet werden, dass die Abtastgeschwindigkeit durchgehend gleich bleibt. Jedoch ist der Beschleunigungssensor beim Messen zuverlässig und kann zu jeder Zeit verwendet werden. Dagegen kann das *Global Positioning System* (GPS) ausfallen z.B. wenn sich das Smartphone in einem Gebäude befindet. Der Beschleunigungssensor verbraucht im Gegensatz zu anderen Bewegungssensoren 10-mal weniger Energie.

Gyroskop

Das Gyroskop ist sowie der Beschleunigungssensor als ein hardware-basierter Sensor im Smartphone eingebaut. Es misst die Rate der Rotation des Smartphones um alle drei physischen Achsen. Die Einheit der vom Gyroskop gelieferten Werte ist $\frac{rad}{s}$. Das Gyroskop liefert also die Winkelgeschwindigkeit

um die drei physischen Achsen. Hauptsächlich wird das Gyroskop beim Drehen oder Rotieren des Geräts verwendet.

Das Prinzip, wie die Werte vom Gyroskop erhalten werden, ist identisch mit dem Prinzip beim Beschleunigungssensor. Zuerst wird ein selbst definierter *Sensor Event Listener* beim *Sensor Manager* registriert und die gewünschte Abtastgeschwindigkeit definiert. Bei der Auswahl von der Abtastgeschwindigkeit gilt das Gleiche wie beim Beschleunigungssensor. Die Abtastgeschwindigkeit sollte passend ausgewählt werden.

Vom Gyroskop werden folgende *Sensor Events* erhalten:

Winkelgeschwindigkeit um die X, Y und Z-Achse, der Zeitstempel, die Genauigkeit des *Sensor Events* und der Sensor, von dem das *Sensor Event* geschickt wurde. Hier ist es das Gyroskop.

Der *Sensor Event* vom Gyroskop unterscheidet sich nur in den Wert-Feldern. Diese enthalten beim Gyroskop die Winkelgeschwindigkeit um die physischen Achsen anstatt die Beschleunigung auf den physischen Achsen (Beschleunigungssensor). Je nach Sensor unterscheiden sich nur diese Wert-Felder. Die restlichen Informationen des *Sensor Events* wie der Zeitstempel und die Genauigkeit des *Sensor Events* bleiben gleich.

Hier kann der Zeitstempel auch dafür dienen, die Geschwindigkeit zu messen, wie oft das Gyroskop *Sensor Events* schickt. Aus den gleichen Gründen wie beim Beschleunigungssensor, kann diese Geschwindigkeit nicht direkt abgefragt werden und es ist kein Verlass darauf, dass diese Geschwindigkeit gleich bleibt, was nachteilig ist. Wenn die Zuverlässigkeit und Verfügbarkeit des Gyroskops auch mit dem GPS verglichen wird, ist das Gyroskop, wie der Beschleunigungssensor, zuverlässiger und verfügbarer als das GPS, weil das GPS unter gewissen Umständen ausfallen kann.

Lineare Beschleunigung

Der Sensor, der Werte für die lineare Beschleunigung liefert, kann je nach Smartphone variieren. Es kann entweder hardware-basiert oder software-basiert sein. Im Falle eines software-basierten Sensors für die lineare Beschleunigung werden mittels anderen Sensoren die lineare Beschleunigung gemessen bzw. errechnet. Der Sensor für die lineare Beschleunigung liefert drei Werte, die aus den drei physikalischen Achsen, X, Y und Z ausgelesen werden. Die Erdanziehungskraft ist im Gegensatz zum Beschleunigungssensor in diesen Werten nicht enthalten aber die Einheit ist dieselbe, nämlich $\frac{m}{s^2}$. Ein Anwendungszweck ist zum Beispiel das Monitoring der linearen Beschleunigung entlang einer Achse. Die Relation lautet:

lineare Beschleunigung = Beschleunigung - Erdbeschleunigung

Aus dem *Sensor Event*, das vom Sensor für die lineare Beschleunigung geschickt wird, können folgende Werte ausgelesen werden:

Lineare Beschleunigung auf der X, Y und Z-Achse, der Zeitstempel, die Genauigkeit des *Sensor Events* und der Sensor, von dem das *Sensor Event* geschickt wurde. Hier ist es der Sensor für die lineare Beschleunigung.

Dieser Sensor wird eher dann verwendet, wenn die Erdanziehungskraft außer Acht gelassen werden soll. So muss beispielsweise im Vergleich zum Beschleunigungssensor die Erdanziehungskraft nicht manuell herausgefiltert werden. Es kann auch dafür verwendet werden, die Geschwindigkeit vom Auto zu messen. Jedoch muss davor mit einem kleinen Mehraufwand vor der Messung ein Kalibrierungsschritt durchgeführt werden, weil der Sensor für die lineare Beschleunigung immer einen Offset hat.

Weitere Sensorik

Wie in Tabelle 2.1 zu sehen ist, gibt es noch viele weitere Sensoren, mit denen ein Android-Smartphone ausgestattet ist. In diesem Kapitel werden einige weitere Sensoren näher erläutert und beschrieben.

Neben dem Beschleunigungssensor, Gyroskop und dem Sensor für die lineare Beschleunigung gibt es weitere Sensoren, die zum Beispiel die Raumtemperatur, die Temperatur des Geräts, die Intensität des Umgebungslichts und den Luftdruck der Umgebung angeben. Die Einheit der Intensität des Umgebungslichts ist in Lux (lx). Der Sensor, der die Intensität des Umgebungslichts misst, wird beispielsweise dazu verwendet, die Bildschirmhelligkeit passend der Umgebungsbeleuchtung zu regulieren. Der Annäherungssensor misst die Entfernung zu einem Objekt vom Bildschirm des Geräts aus in Zentimeter. Dieser Sensor wird typischerweise dafür verwendet, um zu erkennen, ob das Smartphone vom Benutzer an sein Ohr gehalten wird. Dies ist beim Telefonieren der Fall. Weitere Sensoren sind aus der Tabelle 2.1 zu entnehmen.

2.2 Research in the Large

Im Bereich der Mensch-Computer-Interaktion hat die Wichtigkeit der kontrollierten Testumgebung zugenommen. Dabei kann es sich um diverse Studien handeln, die zum Beispiel neue Interaktionsmethoden testen [6]. Die Durchführung einer derartigen Studie erfolgt üblicherweise mit bestimmten Teilnehmern aus einer speziellen Bevölkerungsgruppe. Häufig werden für diese Studien eigene Studenten an Universitäten eingesetzt. Diese werden von jeweiligen Forschern kontrolliert durchgeführt. Es werden reproduzierbare Umgebungen erstellt, wo der Forscher allmögliche relevanten Faktoren beeinflussen kann und so die interne Validität gesichert wird. Jedoch stellt sich bei einer derartigen Studie die Frage, ob die Feststellungen aus der Studie verallgemeinert werden können. Denn in den meisten Fällen kann kein Zusammenhang zwischen dem Benutzer und den Ergebnissen der Studie gebildet werden. [5, 11]

Die Forschungsmethode *Research in the Large* hilft, das Ergebnis der Studie im Zusammenhang mit Benutzern zu verallgemeinern. Hierbei handelt es sich um eine Umgebung bei der, im Vergleich zur oben genannten Durchführung, weitaus mehrere Teilnehmer involviert sind. Die Studien im Bereich der Mensch-Computer-Interaktion ändern sich mit dem starken Wachstum des *mobile computings*. Der Desktop-Rechner wird immer mehr von mobilen Endgeräten wie Smartphones, Tablets und Laptops verdrängt. Diese mobilen Endgeräte werden überall in sehr vielen Umgebungen und unterwegs verwendet. Forscher der Mensch-Computer-Interaktion setzen allgegenwärtige Computertechnologien in Studien ein. Die Feststellungen und Ergebnisse dieser Studien sind im Vergleich zu den traditionellen Studien, die in einer kleineren Umgebung durch den Forscher vor Ort durchgeführt werden, repräsentativer. Dadurch kann aber die interne Validität der neuartigen Studie niedriger als die interne Validität der traditionellen Studie sein. Jedoch sind die Ergebnisse repräsentativer als die Ergebnisse aus der traditionellen Studie. [11]

Im Idealfall sollte eine Studie dergleichen in allmöglichen Umständen und mit zahlreichen Benutzerklassen durchgeführt werden. Bei diesen Umständen handelt es sich um Umstände, in denen das System auch verwendet wird. Die Verwendung des mobilen Endgeräts in allen möglichen Zusammenhängen zu berücksichtigen, ist nicht einfach. Durch die App Marktplätze wie *Google Play* von Google oder *AppStore* von Apple, die als sehr weit verteilte Kanäle für App's gelten und mobile Anwendungen (Apps) anbieten, haben die Forscher die Möglichkeit, sehr viele Studienteilnehmer für Mensch-Computer-Interaktion-Studien zu erreichen [5]. Den Forschern wird durch das Veröffentlichlichen von Apps in den App Marktplätzen die Erfassung von großen Datenmengen in seiner natürlichen Umgebung zu untersuchen ermöglicht. [5, 11]

Projekte, die durch *Research-in-the-Large* umgesetzt wurden

In diesem Abschnitt werden Projekte und Arbeiten vorgestellt, die mit der Forschungsmethode *Research in the Large* umgesetzt wurden. Das erste Beispiel handelt von *SmudgeSafe*. In *SmudgeSafe* geht es um die Eingabe von grafischen Passwörtern. Hier wurden zwei Nutzerstudien durchgeführt. In der kleinen Studie haben 20 Teilnehmer versucht, benutzerdefinierte Passwörter herauszufinden.

In der großen Studie haben 374 Teilnehmer an der Studie teilgenommen und 130.000 Anmeldeinformationen generiert. [12]

Ein weiteres Beispiel ist das Spiel *Hit It!*, eine Android App, in der Kreise angezeigt werden, die der Nutzer anklicken soll. Es handelt sich um die Analyse des Touchverhaltens von Smartphones. In der Studie wurden mehr als 120 Mio. Daten bei 91.731 Installationen gesammelt. *Hit It!* wurde dafür entwickelt und veröffentlicht, um diese Daten für die Analyse zu sammeln. [13]

Type It! ist mit der Absicht entwickelt und veröffentlicht worden, viele Tastenanschläge auf virtuellen Tastaturen von Smartphones zu sammeln. Es ist ein Spiel, indem Wörter angezeigt werden, die der Benutzer eintippen muss. Durch *Type It!* wurden insgesamt 47.770.625 Tastenanschläge bei 72.945 Installationen gesammelt. [14]

In den letztgenannten Beispielen wurden die Daten mithilfe einer spielartigen Android App gesammelt. Die gemeinsame Problematik bei solchen Studien ist, dass mit 0 Teilnehmern begonnen wird. Jedoch ist die Wahrscheinlichkeit, dass viele Nutzer an der Studie teilnehmen groß. Der Mehrwert für den Nutzer ist hoch, da die App in Form eines Spiels umgesetzt wird.

Sensor	Type	Description
TYPE_ACCELEROMETER	Hardware	Measures the acceleration force in m/s ² that is applied to a device on all three physical axes (x, y, and z), including the force of gravity.
TYPE_AMBIENT_TEMPERATURE	Hardware	Measures the ambient room temperature in degrees Celsius (°C). See note below.
TYPE_GRAVITY	Software or Hardware	Measures the force of gravity in m/s ² that is applied to a device on all three physical axes (x, y, z).
TYPE_GYROSCOPE	Hardware	Measures a device's rate of rotation in rad/s around each of the three physical axes (x, y, and z).
TYPE_LIGHT	Hardware	Measures the ambient light level (illumination) in lx.
TYPE_LINEAR_ACCELERATION	Software or Hardware	Measures the acceleration force in m/s ² that is applied to a device on all three physical axes (x, y, and z), excluding the force of gravity.
TYPE_MAGNETIC_FIELD	Hardware	Measures the ambient geomagnetic field for all three physical axes (x, y, z) in μT .
TYPE_ORIENTATION	Software	Measures degrees of rotation that a device makes around all three physical axes (x, y, z). As of API level 3 you can obtain the inclination matrix and rotation matrix for a device by using the gravity sensor and the geomagnetic field sensor in conjunction with the <code>getRotationMatrix()</code> method.
TYPE_PRESSURE	Hardware	Measures the ambient air pressure in hPa or mbar.
TYPE_PROXIMITY	Hardware	Measures the proximity of an object in cm relative to the view screen of a device. This sensor is typically used to determine whether a handset is being held up to a person's ear.
TYPE_RELATIVE_HUMIDITY	Hardware	Measures the relative ambient humidity in percent (%).
TYPE_ROTATION_VECTOR	Software or Hardware	Measures the orientation of a device by providing the three elements of the device's rotation vector.
TYPE_TEMPERATURE	Hardware	Measures the temperature of the device in degrees Celsius (°C). This sensor implementation varies across devices and this sensor was replaced with the TYPE_AMBIENT_TEMPERATURE sensor in API Level 14

Tabelle 2.1: Sensorübersicht [10]

3 Das Konzept

In diesem Kapitel wird die konzeptionelle Umsetzung der nutzerzentrierten Datenerhebung zur Aktivitätserkennung beschrieben. *Acremode* ist ein Spiel, das aus vier unterschiedlichen Spielen besteht. Bei den Spielen handelt es sich jeweils um Herausforderungen an einen zweiten Spieler. Es ist also in dem Sinne ein Multiplayer-Spiel. Der zweite Spieler kann entweder ein Freund oder ein von *Acremode* zufällig ausgewählter Spieler sein. Der herausfordernde Spieler wählt eins von den vier Spielen, spielt das Spiel und fordert damit den zweiten Spieler heraus, der das gleiche Spiel wie der herausfordernde Spieler spielen muss. Derjenige, der das bessere Ergebnis erreicht, gewinnt das Spiel. *Acremode* besteht aus folgenden vier Spielen:

- Long Distance
- Typing Speed
- Reaction Time
- Shake Me

Long Distance

In diesem Spiel muss der Benutzer eine lange Distanz durch das Laufen erreichen. Für die Messung der hinterlegten Distanz wird der GPS-Sensor verwendet. Es wird in einem bestimmten Intervall das aktuelle *Location*-Objekt gespeichert. Nach dem Speichern wird die Distanz zwischen dem aktuell gespeicherten und davor gespeichertem Ort berechnet und auf die Gesamtdistanz addiert. Die Spieldauer beträgt eine Minute. Wer von den beiden Spielern eine längere Distanz erreicht, gewinnt das Spiel.

Bei *Long Distance* werden folgende Daten erfasst:

- Beschleunigung auf allen 3 Achsen (Beschleunigungssensor)
- Nähe eines Objekts zur Frontseite des Geräts in cm (Annäherungssensor)
- Beleuchtungsstärke in Lux (Lichtsensor)
- Liste mit den Distanzen, die alle 2 Sekunden erfasst wurden

Typing Speed

In *Typing Speed* wird die Tippgeschwindigkeit des Spielers gemessen. Als Messeinheit wird *cw/min*, also die Anzahl der korrekt eingegebenen Wörter pro Minute, verwendet. Die Spieldauer ist bei diesem Spiel 30 Sekunden. Es werden aus einem vordefinierten Wörterbuch, welches aus einzelnen englischen Wörtern besteht, zufällig ausgewählte Wörter angezeigt. Bei der richtigen Eingabe des Wortes erscheint das nächste zufällig ausgewählte Wort. Der Spieler mit der schnelleren Tippgeschwindigkeit gewinnt das Spiel.

In diesem Spiel werden folgende Daten erfasst:

- Beschleunigung auf allen 3 Achsen (Beschleunigungssensor)
- Nähe eines Objekts zur Frontseite des Geräts in cm (Annäherungssensor)
- Beleuchtungsstärke in Lux (Lichtsensor)
- Liste mit den korrekt eingegebenen Wörtern

Reaction Time

In *Reaction Time* wird die Reaktionsgeschwindigkeit des Spielers gemessen. Es erscheint zunächst eine Textbox mit der Information, wie der Spielablauf ist. Der Benutzer muss den 3-Sekunden-Countdown mit einem Klick auf den gegebenen Button starten. Nach 3 Sekunden wird der Hintergrund rot und nach einer zufälligen Zeitspanne, die zwischen 0 und 5 Sekunden liegt, wechselt die Farbe des Hintergrunds auf grün. Bei grün muss der Spieler so schnell wie möglich auf den Button klicken. Aus Fairness Gründen wird der Spieler bestraft, wenn er bei rot auf den Button klickt. In diesem Fall wird der Reaktionszeit 0,5 Sekunden je falschem Klick dazu addiert.

Die Reaktionszeit wird allgemein, ausgenommen der Bestrafung, wie folgt berechnet:

$$\text{Reaktionszeit} = \text{ZeitbeiKlick} - \text{ZeitbeiWechselaufgrün} = \text{Ergebnis}$$

Im Falle einer Bestrafung sieht die Berechnung wie folgt aus:

$$\text{Reaktionszeit} = (\text{ZeitbeiKlick} - \text{ZeitbeiWechselaufgrün})$$

$$\text{Ergebnis} = \text{Reaktionszeit} + (F * 0.5)$$

F ist die Anzahl der Klicks bei rot, die einzeln mit 0.5 Sekunden bestraft werden.

Der Spieler mit der kürzeren Reaktionszeit gewinnt das Spiel. Die Datenerhebung beginnt bei *Reaction Time* mit dem ersten Klick auf den Countdown-Button und endet mit dem abschließenden Klick bei grün.

Dabei werden folgende Daten erfasst:

- Beschleunigung auf allen 3 Achsen (Beschleunigungssensor)

-
- Nähe eines Objekts zur Frontseite des Geräts in cm (Annäherungssensor)
 - Beleuchtungsstärke in Lux (Lichtsensoren)

Shake Me

Die Spieler haben in *Shake Me* 10 Sekunden Zeit, das Smartphone so schnell wie möglich zu schütteln. In diesen 10 Sekunden wird die schnellste Beschleunigung als das Ergebnis des Spiels festgehalten. Der Spieler, der eine größere Beschleunigung erreicht, gewinnt das Spiel. Das Smartphone muss also eine hohe Beschleunigung in diesen 10 Sekunden erfahren. Die Einheit bei *Shake Me* ist $\frac{m}{s^2}$. In 10 Sekunden werden folgende Daten erfasst:

- Beschleunigung auf allen 3 Achsen (Beschleunigungssensor)
- Nähe eines Objekts zur Frontseite des Geräts in cm (Annäherungssensor)
- Beleuchtungsstärke in Lux (Lichtsensoren)

4 Acremode

In diesem Kapitel wird die Android App *Acremode* behandelt. Zuerst wird in Abschnitt 4.1 die Android App *Acremode*, in Abschnitt 4.2 die Architektur und in Abschnitt 4.3 die Spezifikation von *Acremode* beschrieben. *Acremode* wurde im Rahmen dieser Arbeit entwickelt und für die nutzerzentrierte Datenerhebung zur Aktivitätserkennung verwendet. Die App wurde in *Google Play* veröffentlicht und kann von Android-Nutzern installiert und benutzt werden ¹.

4.1 Beschreibung

Es wird zunächst der Ablauf der Erhebung von Daten in Kapitel 4.1.1 beschrieben. Anschließend werden die benutzten Sensoren (Kapitel 4.1.2) näher erläutert. Auf die erhobenen Daten wird in Kapitel 4.1.3 eingegangen. Das Gesamtkonzept basiert auf einer spielartigen Android App. Es werden Sensordaten bei bestimmten Aktivitäten gesammelt.

4.1.1 Ablauf der Erhebung von Daten

In Abbildung 4.1 wird der Ablauf der Datenerhebung dargestellt. Es handelt sich um ein Spiel mit vier verschiedenen Spielen, die vom Benutzer ausgewählt und innerhalb der gegebenen Zeitdauer gespielt werden. Das "Spiel" in Abbildung 4.1 bezieht sich auf eines der vier Spiele. Zuerst wird ein Spiel gestartet. Wenn das Spiel läuft, beginnt das Erheben von Sensordaten, die im Smartphone zwischengespeichert werden. Welche Daten erhoben werden, wird in Kapitel 4.1.3 und welche Sensoren verwendet werden, wird in Kapitel 4.1.2 beschrieben. Nach der gegebenen Zeitdauer wird das Spiel beendet und anschließend die zwischengespeicherten Daten archiviert. Das Archiv enthält so viele Listen wie die Anzahl der Sensoren, die bei der Datenerhebung verwendet wurden. Für jeden Sensor wird eine Liste mit den erhobenen Sensordaten erstellt. Diese Listen werden in einer Datei archiviert. Die archivierte Datei wird folglich auf den Server hochgeladen.

Auf dem Server werden alle hochgeladenen Dateien nach den Spielen kategorisiert gespeichert. Die Dateien werden wie folgt benannt: *YYYY-MM-TT-HH-MM-SS-X-archive.zip*. *X* steht für das Spiel, das während der Datenerhebung gespielt wurde.

¹Acremode: <https://play.google.com/store/apps/details?id=com.halilbugdayci.acremode> [accessed 01-October-2014]

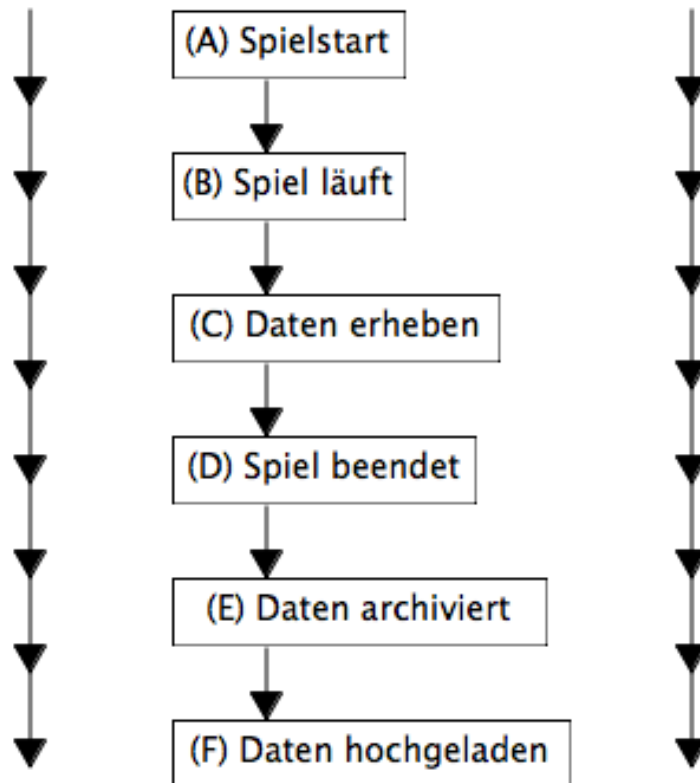


Abbildung 4.1: Ablauf der Datenerhebung

4.1.2 Benutzte Sensoren

In Kapitel 2.1 wurden einige unterschiedliche Sensoren beschrieben, die in einem Android Smartphone integriert sind und für die Aktivitätserkennung verwendet werden können. Folgende Sensoren sind bei dieser Arbeit zum Einsatz gekommen:

- Beschleunigungssensor
- GPS
- Annäherungssensor
- Lichtsensor

Das GPS wird nur in einem von vier Spielen, in *Long Distance*, verwendet. Die restlichen Sensoren werden in allen der vier Spiele eingesetzt. In Kapitel 4.1.3 werden die genauen Details über die erhobenen Sensordaten beschrieben. Die Abtastgeschwindigkeit der Sensoren wurde bereits in Kapitel 2.1 näher erläutert.

4.1.3 Erhobene Daten

Dieser Abschnitt behandelt die erhobenen Sensordaten. Alle erhobenen Daten sind anonym und auf keine Person zurückzuführen. Für jeden Sensor, von dem die Sensordaten erhoben werden, wird eine kommaseparierte Liste (.csv) erstellt. Diese Liste ist abhängig vom Sensor in unterschiedliche Spalten unterteilt.

Die Liste beim Beschleunigungssensor besteht aus folgenden Spalten:

- Zeitstempel (timestamp)
- Beschleunigung auf der X-Achse (acceleration on X-axis)
- Beschleunigung auf der Y-Achse (acceleration on Y-axis)
- Beschleunigung auf der Z-Achse (acceleration on Z-axis)
- Genauigkeit der Angabe (accuracy)

Annäherungssensor:

- Zeitstempel (timestamp)
- Nähe in cm (proximity distance in cm)
- Genauigkeit der Angabe (accuracy)

Lichtsensor:

- Zeitstempel (timestamp)
- Beleuchtungsstärke in Lux (illuminance in lux)
- Genauigkeit der Angabe (accuracy)

Beim GPS-Sensor werden nur die Distanzen in einem Zeitintervall von zwei Sekunden aufgezeichnet:

- Dauer (duration)
- Distanz in m (distance in meters)

Es werden also keine Ortungsdaten aufgezeichnet, die auf eine Person zurückführen können. In dieser Arbeit werden alle benötigten Werte aus den *Sensor Events*, die vom *Sensor Manager* erhalten werden, aufgezeichnet. Die *Sensor Events* werden vom *Sensor Manager* geschickt, wenn sich die Werte vom Sensor ändern. Beim Beschleunigungssensor ändern sich die Werte viel häufiger im Vergleich zum Licht- oder Annäherungssensor, weshalb die Liste mit den Werten aus dem Beschleunigungssensor mehr Einträge als die Liste mit den Werten aus dem Licht- oder Annäherungssensor hat.

4.2 Architektur

Die Architektur, die bei der Umsetzung von *Acremode* verwendet wurde, ist in Abbildung 4.2 zu sehen. Es ist eine Client-Server-Architektur. Der Client ist in diesem Fall *Acremode* und auf dem Webserver sind *PHP* und *MySQL* installiert und eingerichtet. Webserver sind darauf spezialisiert, Anfragen vom Client, zu beantworten. Der Webserver ist für den Client über das Internet zu erreichen. Die Anfragen werden von *Acremode* an den Webserver geschickt. Im Webserver wird die Anfrage bearbeitet und dem Client geantwortet. Anschließend wird die Antwort vom Client *Acremode* verarbeitet.

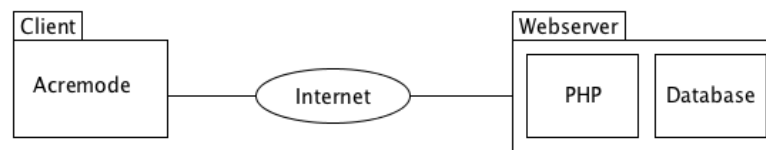


Abbildung 4.2: Architektur von *Acremode*

4.3 Spezifikation

In diesem Kapitel wird die Benutzeroberfläche von *Acremode* genauer beschrieben. Zur Hervorhebung von Begriffen im Abschnitt 4.3.1 werden die folgenden Begriffe verwendet:

Namen und grafisch dargestellte Felder der App werden *kursiv* hervorgehoben.

Die Buttons werden mit einem Rahmen um den Text, der auf dem Button steht, hervorgehoben.

4.3.1 Benutzeroberfläche

Dieser Abschnitt behandelt die gesamte Benutzeroberfläche von *Acremode*. In Abbildung 4.3 sind die ersten Schritte vor der Verwendung von *Acremode* zu sehen. Beim ersten Start wird eine Meldung angezeigt, die in Abbildung 4.3a zu sehen ist. Die Meldung informiert den Benutzer darüber, dass Sensordaten erhoben werden. Unter dieser Meldung ist ein Okay-Button, der zur Bestätigung der Meldung dient. M.Pielot et al. haben mit 4 unterschiedlichen Varianten dieser Meldungen experimentiert, unter denen die in 4.3a zu sehende Meldung sich als die erfolgreichste Meldungsart erwies [15]. Nach der Meldung ist die Anmeldeansicht (Abbildung 4.3b) zu sehen. In dieser Ansicht muss der Benutzer seinen Benutzernamen eingeben und auf Register klicken um fortzufahren. Anschließend hat der Benutzer die Möglichkeit, sein Geburtsdatum anzugeben, die er überspringen und damit direkt zur Begrüßungsansicht gelangen kann. Im letzten Schritt vor der Verwendung von *Acremode* kann der Benutzer *Acremode* mit seinen Freunden teilen, bevor er auf Continue klickt, um auf die Hauptansicht (Abbildung 4.4) der App zu gelangen.

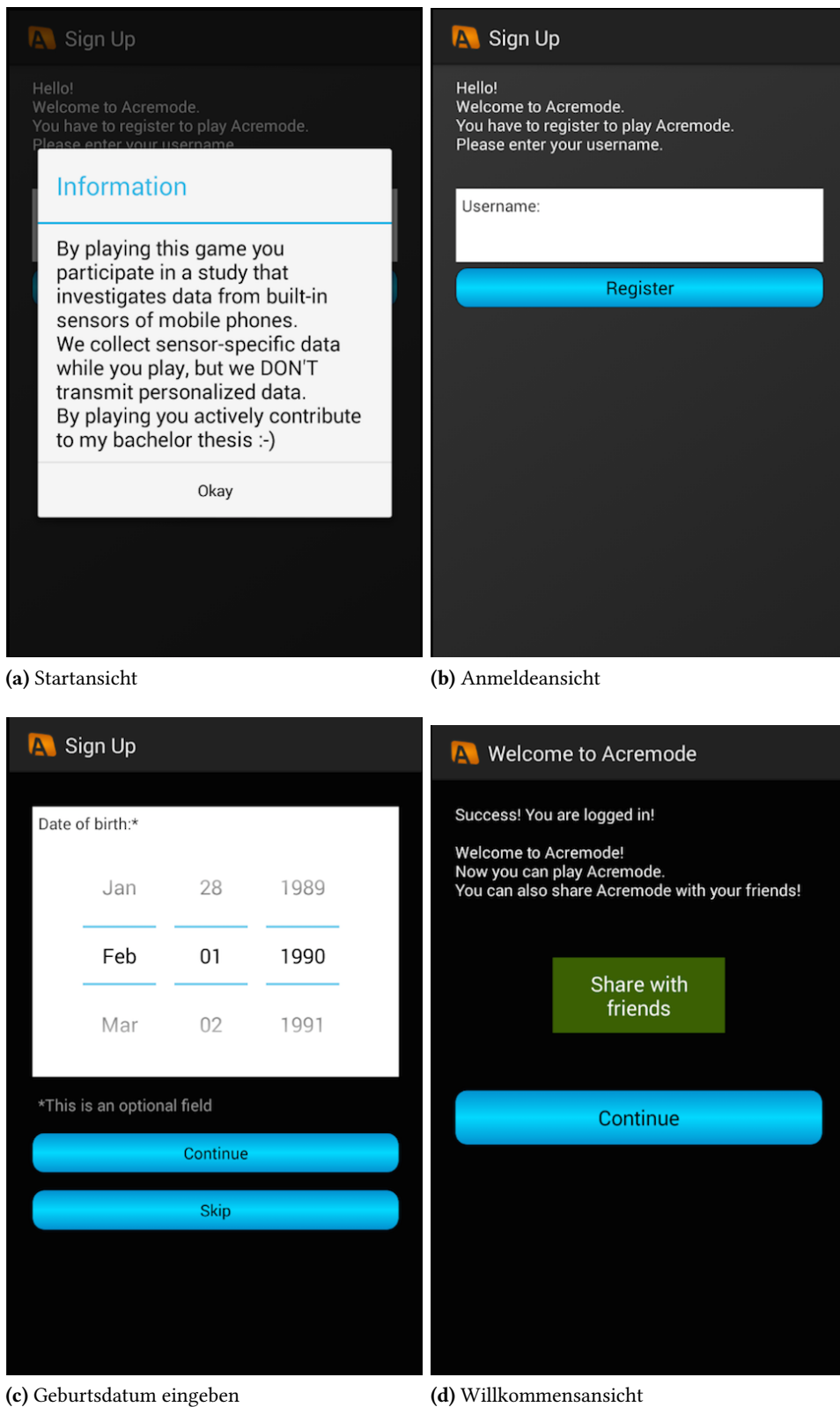


Abbildung 4.3: Die ersten Schritte vor der Verwendung von *Acremode*

4 Acemode

In der Hauptansicht (Abbildung 4.4) sind vier Buttons zu sehen: **New Challenge**, **Requests**, **Game History** und **Friends**. Über diesen Buttons sieht der Benutzer seinen Namen, die Anzahl der Spiele, die Anzahl der gewonnenen Spiele, die Gewinnserie und den Gewinnprozentsatz. Diese sind untereinander aufgelistet.

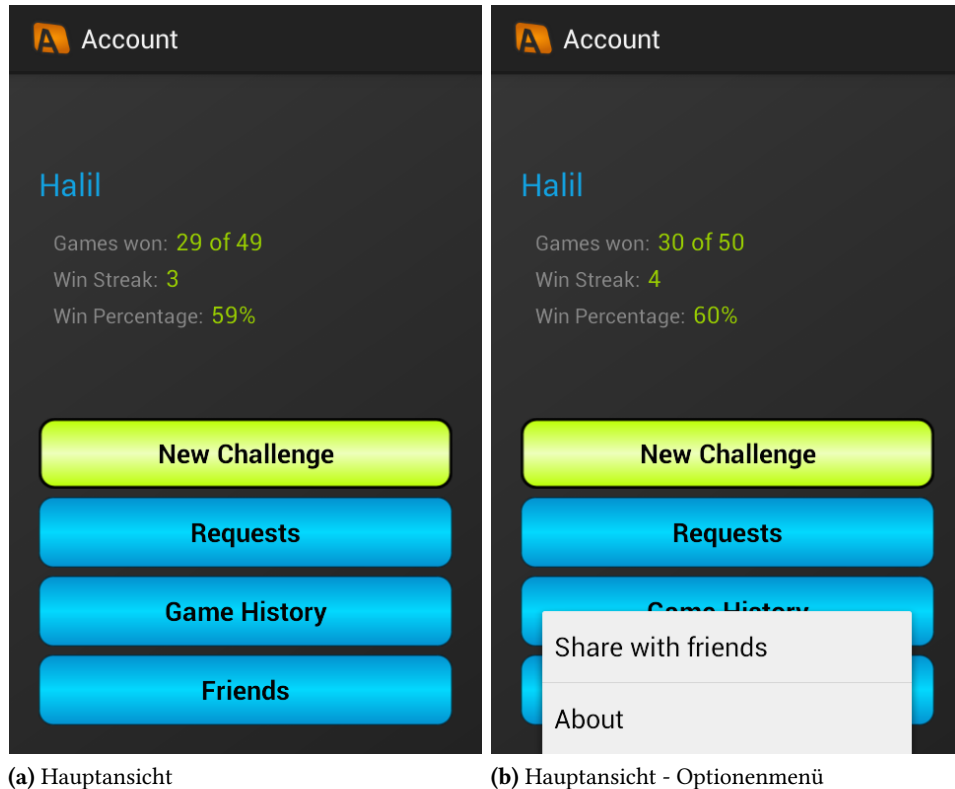


Abbildung 4.4: Hauptansicht von *Acemode*

Beim Klick auf **New Challenge** erscheint eine Liste mit Freunden und einem **Random Player**-Button (Abbildung 4.5a). Nach dem Klick auf einen Spieler oder auf **Random Player** erscheint ein Bestätigungsdialog, der in Abbildung 4.5b abgebildet ist. Falls der Benutzer es bestätigt, erscheint die Ansicht aus Abbildung 4.5c mit den Spielerinformationen des ausgewählten Spielers und vier Buttons für die Spielauswahl. Die Spielabläufe der einzelnen Spiele sind in den Abbildungen von 4.6 bis 4.9 zu sehen.

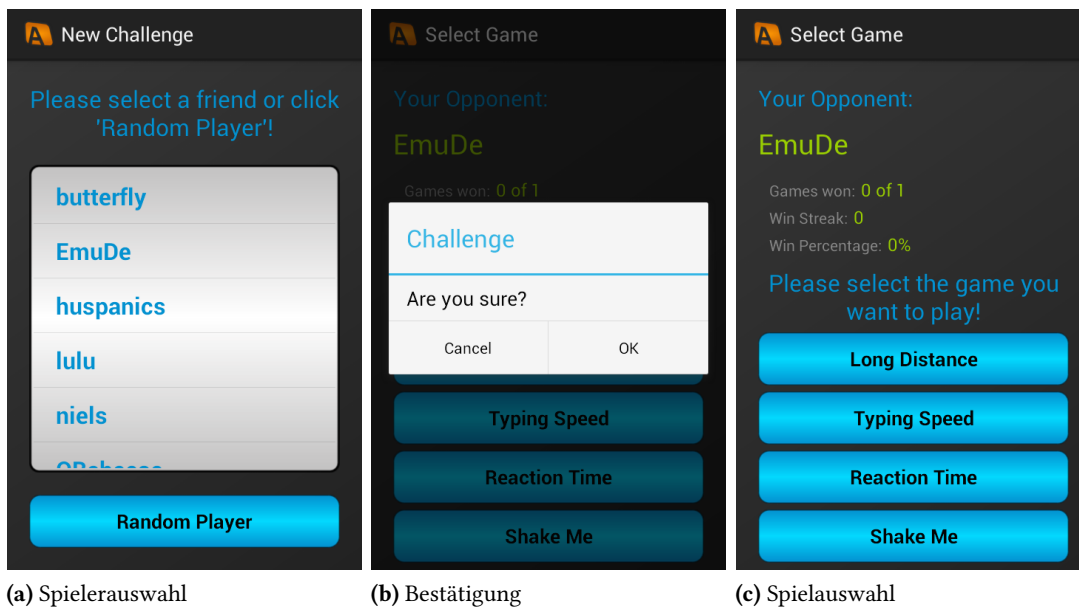


Abbildung 4.5: Der Ablauf, wie der Benutzer einen Gegner herausfordern kann

Long Distance

Wenn der Benutzer *Long Distance* spielen möchte, muss er zu Beginn auf das GPS-Signal warten. Erst wenn ein GPS-Signal vorhanden ist, kann der Benutzer auf `Start!` klicken, um das Spiel *Long Distance* zu starten (Abbildung 4.6b und 4.6c). Wenn das GPS deaktiviert ist, erscheint eine Meldung (Abbildung 4.6a), dass das GPS deaktiviert ist und ein Button `Goto Settings Page to enable GPS`. Mit dem Klick auf `Goto Settings Page to enable GPS` gelangt der Benutzer in das Einstellungs Menü, wo der Benutzer das GPS aktivieren kann. Nach dem Start läuft die Zeit und die aktuell erreichte Distanz wird zur Laufzeit angezeigt (Abbildung 4.6d). Falls zwischendurch das GPS-Signal schwächer wird oder nicht mehr empfangen werden kann, wird die Zeit gestoppt und es erscheint eine Meldung, die in Abbildung 4.6e zu sehen ist. Hier kann der Benutzer entweder warten oder das Spiel aufgeben (Abbildung 4.6f).

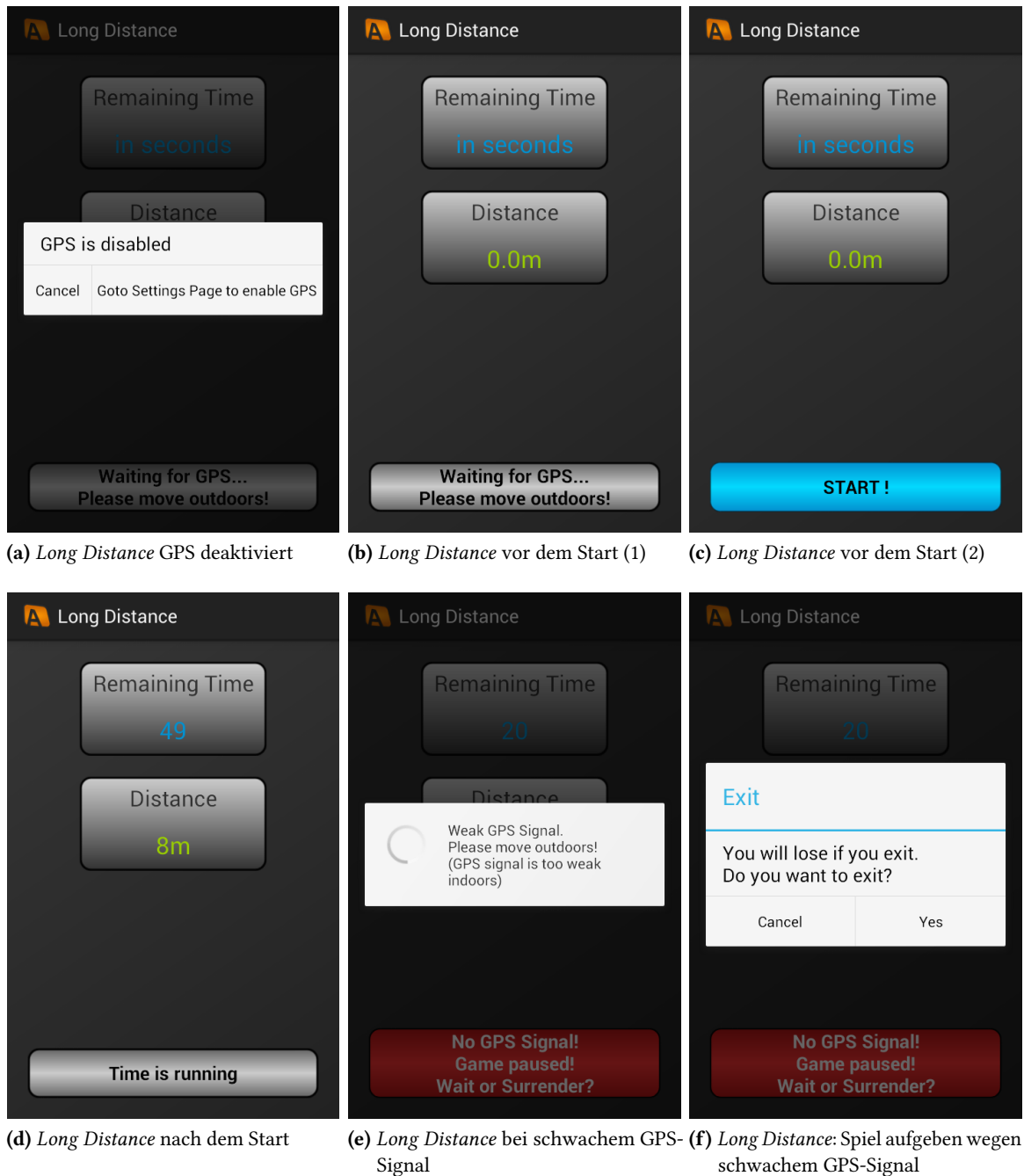


Abbildung 4.6: Spielablauf: *Long Distance*

Typing Speed

Im Spiel *Typing Speed* sieht der Benutzer vor dem Start die Ansicht aus Abbildung 4.7a. Nach einem Klick auf das Textfeld mit dem Inhalt *Click here to start!* startet die Zeit und es wird das einzugebende Wort in der Box *Type this word* (Abbildung 4.7b) angezeigt. Bei richtiger Eingabe des Wortes erscheint das nächste Wort, bis die Zeit abgelaufen ist.

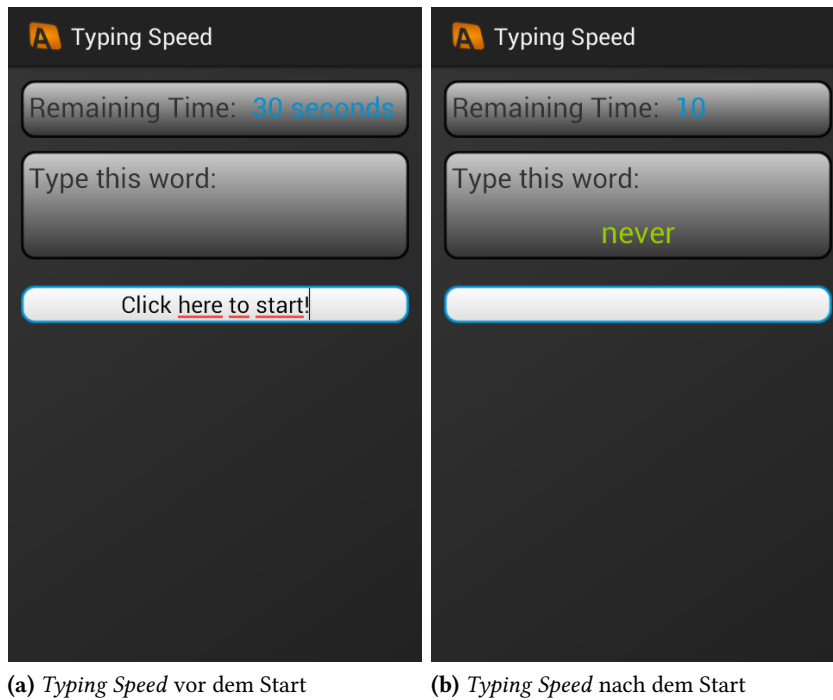


Abbildung 4.7: Spielablauf: *Typing Speed*

Reaction Time

In diesem Spiel wird zuerst eine kurze Beschreibung des Spielablaufs und ein Button `Click here to start countdown!` angezeigt (Abbildung 4.8a). Nach dem Klick auf `Click here to start countdown!` startet der 3-Sekunden-Countdown (Abbildung 4.8b). Anschließend erscheint die Ansicht aus Abbildung 4.8c. Ab jetzt ist der `CLICK, when GREEN!`-Button zu sehen, der bei grün (siehe Abbildung 4.8d) geklickt werden soll.



Abbildung 4.8: Spielablauf: *Reaction Time*

Shake Me

Bei *Shake Me* erscheint zuerst die Ansicht aus Abbildung 4.9a und nach dem Klick auf **START!** läuft die Zeit. Die aktuelle Beschleunigung des Geräts und die erreichte Höchstbeschleunigung wird angezeigt (Abbildung 4.9b).

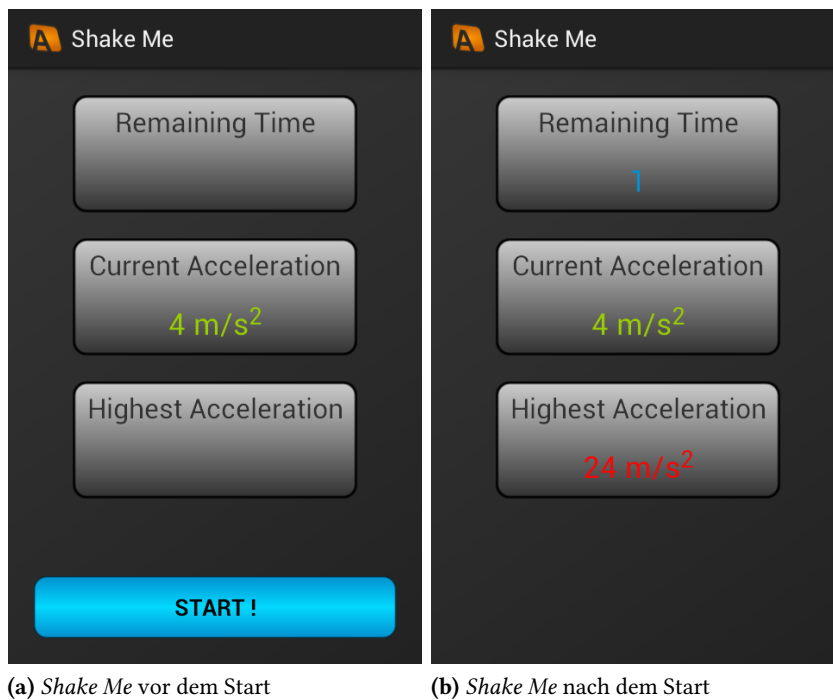
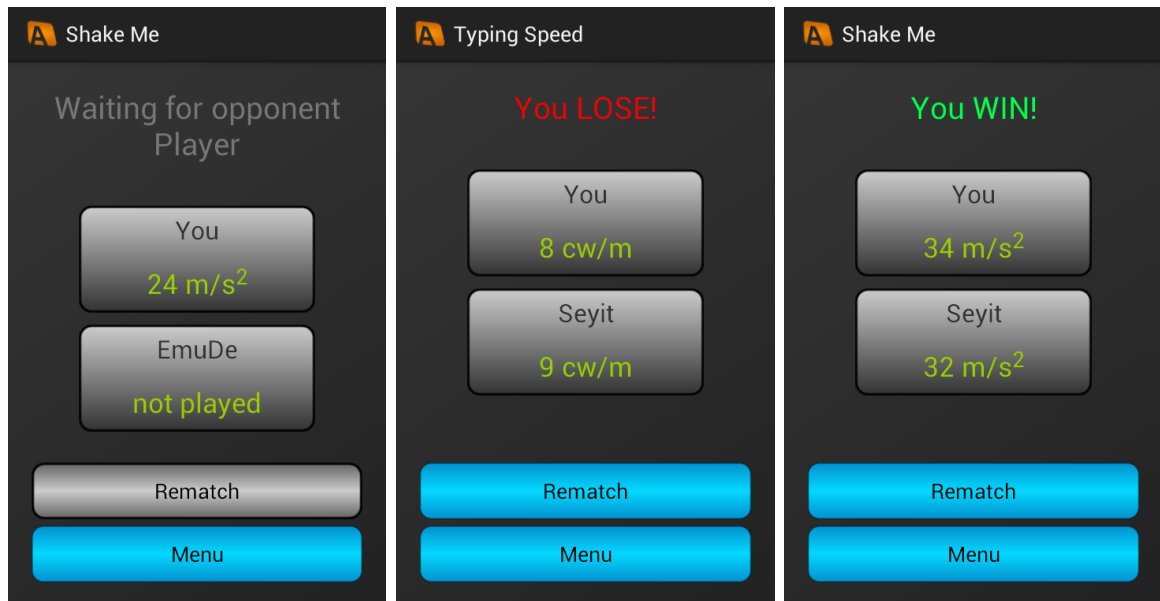


Abbildung 4.9: Spielablauf: *Shake Me*

Nach dem Spiel

Die Ergebnisansicht sieht bei allen Spielen, bis auf die Einheit der Ergebnisse, gleich aus. Wenn der Gegner noch nicht gespielt hat, sieht der Benutzer nur das eigene Ergebnis (siehe Abbildung 4.10a). Wenn der Gegner bereits gespielt hat, wird nach dem Spiel angezeigt, ob der Benutzer gewonnen oder verloren hat (Abbildung 4.10b und 4.10c). Zusätzlich hat der Benutzer die Möglichkeit, das gleiche Spiel mit **Rematch** zu wiederholen, wenn der Gegner bereits gespielt hat.



(a) Gegner hat noch nicht gespielt

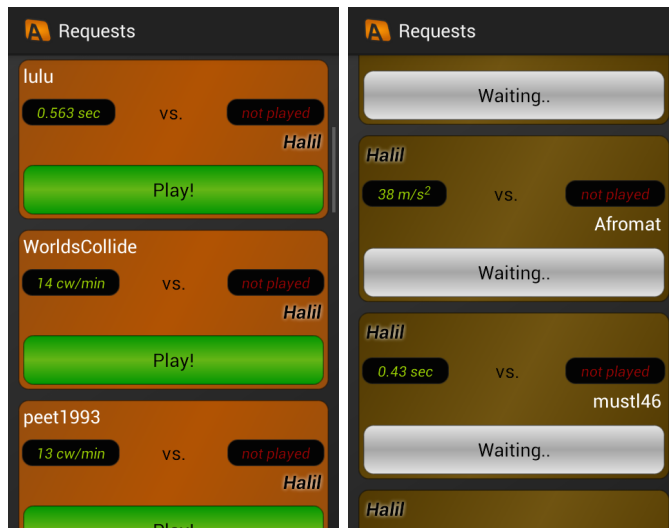
(b) Gegner hat gespielt: verloren

(c) Gegner hat gespielt: gewonnen

Abbildung 4.10: Spielende

Requests

Beim Klick auf `Requests` in Abbildung 4.4a werden alle offenen Herausforderungen angezeigt (Abbildung 4.11a und 4.11b). Wenn der Benutzer Anfragen von Gegenspielern erhält, sieht der Benutzer den `Play!`-Button. Wenn der Benutzer einen Gegenspieler herausgefordert hat und dieser noch nicht gespielt hat, ist dies auch eine offene Anfrage, jedoch sieht der Benutzer in diesem Fall den grau hinterlegten `Waiting..`-Button. Zusätzlich sind die Namen und das Ergebnis der beiden Spieler zu sehen. Der Herausforderer ist immer auf der linken und der Herausgeforderte auf der rechten Seite der Liste zu sehen.



(a) Offene Anfragen (eigene) (b) Offene Anfragen

Abbildung 4.11: Spielanfragen

Game History

Alle offenen und abgeschlossene Spiele erscheinen beim Klick auf `Game History` in Abbildung 4.4a. Die gewonnenen Spiele sind die grünen Boxen und die verlorenen Spiele sind die roten Boxen (Abbildung 4.12).

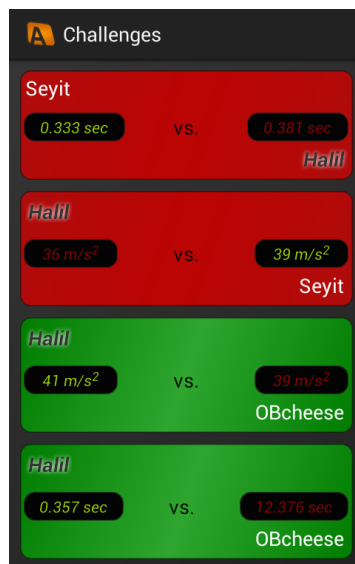


Abbildung 4.12: Game History

Friends

Zur Freundeverwaltung gelangt der Benutzer mit einem Klick auf **Friends** in Abbildung 4.4a. Es erscheint eine Liste mit den Freunden und zwei Buttons: **Add Friend** und **Friend Requests** (siehe Abbildung 4.13a). Beim Klick auf einen Spielernamen aus der Liste (siehe Abbildung 4.13a) erscheint die Ansicht mit den Informationen über den ausgewählten Spieler und einem **Remove**-Button, zum Entfernen eines Freundes (Abbildung 4.13b).

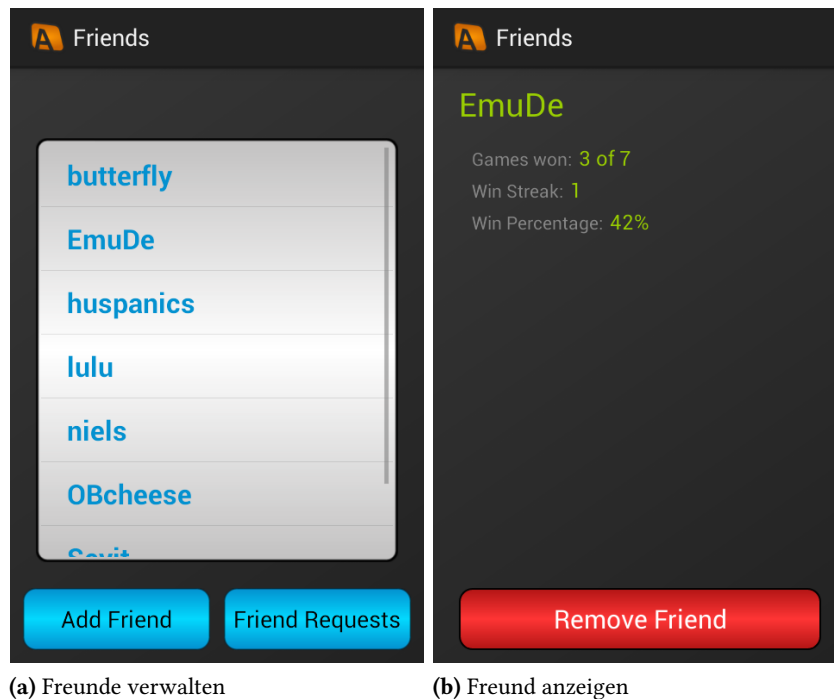


Abbildung 4.13: Ansicht: *Friends*

Mit einem Klick auf **Add Friend** aus Abbildung 4.13a erscheint die Ansicht aus Abbildung 4.14a. In das Textfeld wird der Benutzername vom gesuchten Freund eingegeben und auf **Search!** geklickt. Anschließend erscheint die Liste mit den gefundenen Benutzern, denen per Klick auf den Benutzernamen eine Freundschaftsanfrage verschickt werden kann (siehe Abbildung 4.14b).

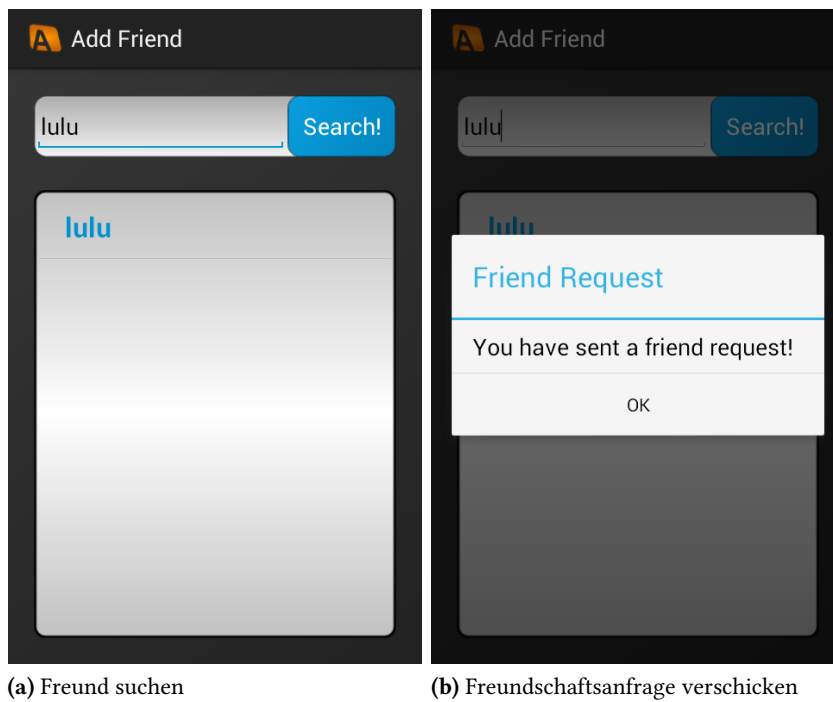
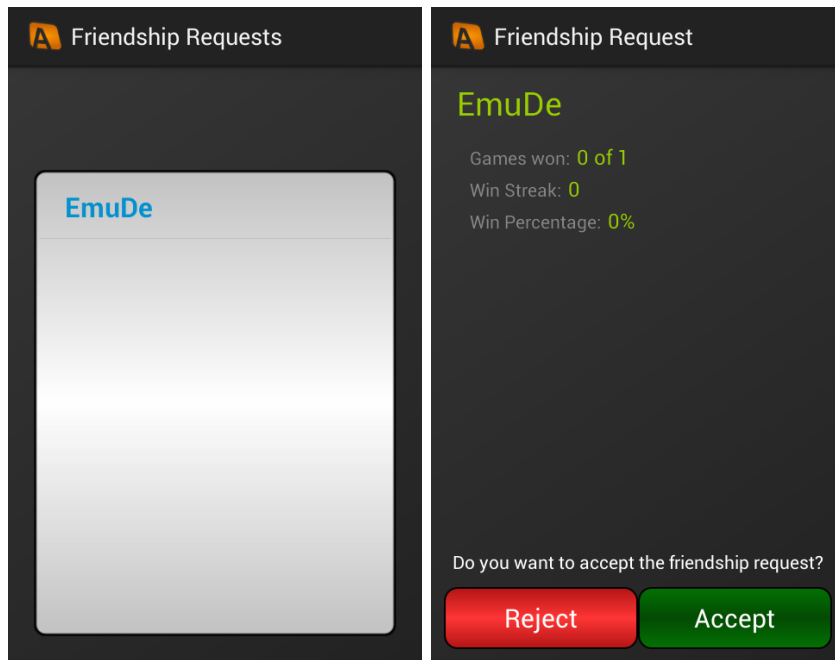


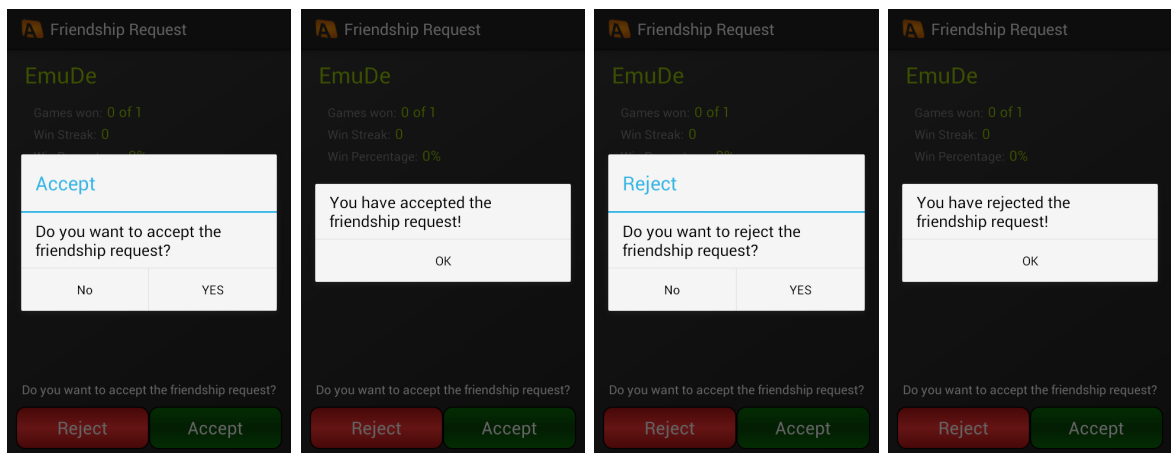
Abbildung 4.14: Freund hinzufügen

Mit einem Klick auf `Friend Requests` aus Abbildung 4.13a erscheint die Ansicht aus Abbildung 4.15a, eine Liste mit nicht bearbeiteten Freundschaftsanfragen. Wird ein Benutzer aus dieser Liste ausgewählt, erscheint die Ansicht aus Abbildung 4.15b. Darunter sind die Buttons `Reject` und `Accept` zu sehen. Bei einem Klick auf einen dieser Buttons, erscheint die Frage, ob der Benutzer die jeweilige Aktion tatsächlich ausführen möchte (siehe Abbildungen 4.15c, 4.15d, 4.15e und 4.15f).



(a) Freundschaftsanfragen

(b) FA anschauen



(c) FA akzeptieren

(d) FA akzeptiert

(e) FA ablehnen

(f) FA abgelehnt

Abbildung 4.15: Freundschaftsanfrage (FA)

5 Studie über Distanzmessung und Nutzerbewertungen

In diesem Kapitel wird die Verlässlichkeit der Distanzmessung (Kapitel 5.1) in *Long Distance* und die Nutzerbewertung (Kapitel 5.2) behandelt. Um die Verlässlichkeit der Distanzmessung des Spiels *Acremode* zu untersuchen, wurde eine Evaluation unter kontrollierten Bedingungen durchgeführt. 10 Probanden liefen eine Distanz von 50m und 100m. Im Folgenden wird zuerst das Design der Studie beschrieben. Anschließend werden die Ergebnisse der Distanzmessung und der Nutzerbewertungen beschrieben und diskutiert.

5.1 Verlässlichkeit der Distanzmessung

In diesem Abschnitt wird die Verlässlichkeit der Distanzmessung des Spiels *Acremode* behandelt. Hierfür wurde eine Evaluation mit 10 Probanden unter kontrollierten Bedingungen durchgeführt. Darunter befanden sich sechs männliche Teilnehmer im Alter von 17-25 und vier weibliche Teilnehmer im Alter von 23-24. Bei den männlichen Teilnehmern handelte es sich um 4 Studenten, 1 Schüler und 1 Azubi. Die weiblichen Teilnehmer bestanden ausschließlich aus Studenten. Zu Beginn der Studie wurde den Teilnehmern *Acremode* vorgestellt und erläutert, dass das Ergebnis der Studie für die Untersuchung der Verlässlichkeit der Distanzmessung verwendet wird. Die notwendige Umgebung, eine Strecke von 50m und 100m, wurde im Freien vorbereitet. Die 50m-Strecke ist in der 100m-Strecke enthalten. Die Aufgabe der Teilnehmer bestand darin, zunächst die 50m-Strecke und anschließend die 100m-Strecke mit dem für die Evaluation bereitgestellten Smartphone zu laufen. Auf dem Android-basierten Smartphone wurde das Spiel vorbereitet, sodass der Teilnehmer nur noch auf den Start-Button klicken und loslaufen musste. Am Ende der Strecke wird die von *Acremode* gemessene Distanz in Metern notiert. Die Dauer der Messung für beide Strecken betrug pro Teilnehmer in etwa zwei Minuten.

Zur Untersuchung der Verlässlichkeit der Distanzmessung wurden insgesamt 20 Läufe durchgeführt. In 10 von 20 diesen Läufen wurden eine Entfernung von 100m gelaufen. In den restlichen 10 Läufen wurden eine Entfernung von 50m gelaufen.

Läufe	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
100m	107m	90m	105m	110m	94m	94m	100m	105m	106m	99m
50m	55m	58m	55m	51m	47m	57m	48m	43m	47m	48m

Tabelle 5.1: 50m und 100m Läufe

In der Tabelle 5.1 ist das Ergebnis der Läufe zu sehen. Sowohl bei den 100m-Läufen als auch bei den 50m-Läufen sind Schwankungen zu erkennen. In der ersten Zeile von Tabelle 5.1 sind die gemessenen Werte der 100m-Strecke aufgelistet. Die Werte liegen teilweise unter 100m und teilweise über 100m, wobei die Standardabweichung bei 5,6m liegt. In der zweiten Zeile sind die gemessenen Werte der 50m-Strecke zu sehen. Auch hier schwanken die Werte und liegen zum Teil unter 50m und zum anderen Teil über 50m. Die Standardabweichung ist in diesem Fall 4,3m. Der Durchschnitt bei den 100m-Läufen ist 101,0m und der Durchschnitt bei den 50m-Läufen ist 50,9m.

Die durchschnittliche Messung der Distanz liefert gute Ergebnisse. Die Standardabweichung zeigt jedoch, dass die Entfernung der einzelnen Messungen von den 100m bzw. 50m abweicht. Desweiteren ist aus der Tabelle 5.1 zu erkennen, dass die maximale Abweichung beim 100m-Lauf 10m ist und beim 50m-Lauf 7m. Ein exakter Wert wurde nur ein einziges Mal gemessen. Insgesamt ist bemerkbar, dass die Messung bei kürzeren Strecken präzisere Werte liefert.

Distanzmessung

Die Distanzmessung wird durch das Zwischenspeichern der *Location*-Objekte umgesetzt. Es wird alle zwei Sekunden das aktuelle *Location*-Objekt gespeichert. Aus dem *hasSpeed*-Feld des *Location*-Objekts wird ausgelesen, ob der Spieler sich bewegt oder steht. Es werden nur die *Location*-Objekte verwendet, die eine Geschwindigkeit (*hasSpeed = true*) haben. Die Gesamtdistanz wird durch das Summieren der Entfernungen zwischen den gespeicherten *Location*-Objekten, die eine Geschwindigkeit haben, berechnet. So werden genauere Messungen erreicht.

5.2 Nutzerbewertung

Acremode wurde von 81 Benutzern installiert. Es haben insgesamt 436 Spiele stattgefunden. Das Spiel, das am meisten gespielt wurde, ist *Reaction Time* mit 197 Spielen, gefolgt von *Typing Speed* mit 114 Spielen, *Shake Me* mit 96 Spielen und *Long Distance* mit 29 Spielen.

Im Folgenden werden die Meinungen der Nutzer über *Acremode* dargestellt. Hierfür haben die Spieler einen Online-Fragebogen ausgefüllt. Es haben 26 Nutzer an der Umfrage teilgenommen. Dabei sollten diese auf die positiven und negativen Aspekte von *Acremode* eingehen und Verbesserungsvorschläge geben.

Unter den positiven Aspekten haben 27% angegeben, dass es amüsant ist mit Freunden zu spielen. Darunter haben 11% angedeutet, dass sie sich gereizt fühlen, wenn sie von Freunden herausgefordert werden und deshalb so oft spielen, bis sie den Gegner übertreffen. Dies sei ein Grund, weshalb sie das Spiel weiterempfehlen. 19% der Bewerter sind der Meinung, dass das Spiel gut gelungen ist und dass es Spaß macht. Weitere 8% finden *Acremode* abwechslungsreich, weil es mehrere verschiedene Spiele beinhaltet. Das Spiel *Typing Speed*, *Skake Me* und *Reaction Time* sind beliebte Spiele bei 11% der Nutzer. Die Beliebtheitsquote des Spiels *Long Distance* liegt bei 7%. Dieses Spiel wurde jedoch auch unter den negativen Aspekten erwähnt. 3% der Nutzer haben angegeben, dass das Spiel *Long Distance* aufwendig ist und sie zu faul sind, es zu spielen. Ein weiterer Nachteil ist laut 26% der Nutzer, dass *Acremode* nur auf Englisch verfügbar ist. Vor allem beim Spiel *Typing Speed* fällt es 15% der Nutzer

schwer, die englischen Wörter abzutippen. 20% finden es nicht gut, dass es keine Benachrichtigung nach dem Spielen des Gegners darüber gibt, wer gewonnen hat. Ein Verbesserungsvorschlag von 15% der Nutzer ist es, eine Liste von den angemeldeten Spielern zu erstellen, sodass es möglich ist weitere Spieler zu sehen und als Freund hinzuzufügen. Außerdem schlagen 12% es vor, mehr Informationen im Spielerprofil zu sehen, wie beispielsweise den Rekord. 7% würden es gut finden, wenn es eine Chatmöglichkeit zwischen den Spielern geben würde. Insgesamt wollen 20% der Nutzer, dass weitere Spiele zu *Acremode* hinzugefügt werden.

Aus den Ergebnissen der Umfrage ist zu sehen, dass Erweiterungsmöglichkeiten vorhanden und von den Nutzern erwünscht sind. Das Hinzufügen von weiteren Spielen würde zu mehr Abwechslung führen und die Nutzer hätten eine größere Auswahl. Um den Spaßfaktor des Spieles zu erhöhen können nicht nur weitere Spiele hinzugefügt werden, sondern auch Verbesserungen und Erweiterungen vorgenommen werden. Die Erweiterung von Benachrichtigungen ist in vieler Hinsicht von Vorteil. Wird nach dem Spiel eine Rückmeldung an die Spieler gegeben, dient die Benachrichtigung als Erinnerung das Spielergebnis einzusehen oder eine neue Herausforderung zu starten. Das heißt, es sollen beide Spieler benachrichtigt werden, wenn ein Spiel beendet wird. Ein weiterer wichtiger Vorschlag für die Erweiterung von *Acremode* ist das Anzeigen von zusätzlichen Informationen im Spielerprofil. Das Anzeigen des maximal erreichten Rekords der Spieler motiviert den Gegner den eigenen Rekord zu steigern. Als letzter Punkt ist zu erwähnen, dass die Standardsprache von *Acremode* Englisch ist und es in keiner weiteren Sprache bedienbar ist. Um mehr Personen zu erreichen können weitere Sprachen hinzugefügt werden. In erster Linie kann im Spiel *Typing Speed* eine Option eingebaut werden, in der die Spieler die Sprache der zu tippenden Wörter auswählen. Abschließend lässt sich sagen, dass an *Acremode* einige Verbesserungen vorgenommen werden können. Dennoch sind die Nutzer insgesamt zufrieden.

6 Auswertung der Sensordaten

Im Rahmen dieser Arbeit wurde die Android App *Acremode* entwickelt, die sehr viele Sensordaten gesammelt hat. In diesem Kapitel werden die Daten vom Beschleunigungssensor ausgewertet. Es kann eine Aussage darüber gemacht werden, in welcher Stellung das Smartphone bei den jeweiligen Spielen liegt oder gehalten wird. Darüber hinaus könnten beispielsweise die Sensordaten, die bei *Long Distance* gesammelt wurden, als Trainingsdaten für bestimmte Algorithmen zur Aktivitätserkennung verwendet werden.

6.1 Ausrichtung des Smartphones

Die Ausrichtung des Smartphones ist durch die Sensordaten vom Beschleunigungssensor feststellbar. In Abbildung 6.1 ist das zum Smartphone relative Koordinatensystem zu sehen, das von der Sensor API in Android verwendet wird.

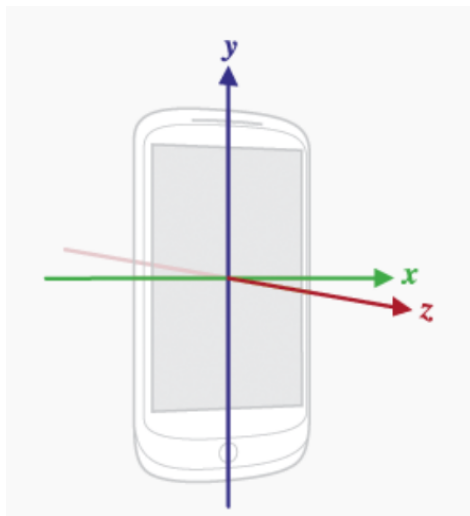


Abbildung 6.1: Koordinatensystem (relativ zum Smartphone) [16]

Die Werte bei einem aufrecht gestellten Smartphone, wie in Abbildung 6.1, sind für die X-Achse 0 , für die Y-Achse $9,8$ (Erdbeschleunigung) und für die Z-Achse 0 . Bei Neigung des Smartphones steigen oder sinken die Werte für die entsprechenden Achsen. Im schnellen Bewegungsmoment des Smartphones sind genaue Angaben über die Koordinatenwerte nicht möglich. Das Bewegen des

Smartphones erschwert die Feststellung der Ausrichtung des Smartphones. Deshalb handelt es sich hierbei um eine ungefähre Ausrichtung des Smartphones.

In allen Spielen bis auf *Shake Me* konnte die durchschnittliche Ausrichtung des Smartphones ermittelt werden. Das Spiel *Shake Me* beabsichtigt das Schütteln, also das schnelle Bewegen des Smartphones, welches die Ermittlung der durchschnittlichen Ausrichtung erschwert und es somit fast unmöglich macht. Die durchschnittliche Ausrichtung des Smartphones ist bei den Spielen *Long Distance*, *Reaction Time* und *Typing Speed* sehr ähnlich. Sie ist wie in Abbildung 6.1 und etwas nach hinten geneigt, sodass das Display des Smartphones weiter nach oben schaut.

6.2 Gesammelte Sensordaten

Acremode wurde in Google Play veröffentlicht und hat in 4 Wochen ca. 1,2 Mio *Sensor Events* gesammelt. Diese wurden auf dem Server archiviert. In Kapitel 4.1.3 wurde beschrieben, welche Sensordaten gesammelt wurden. Die meisten *Sensor Events* wurden im Spiel *Typing Speed* gesammelt - ca. 724.000 *Sensor Events*. Im Spiel *Reaction Time* wurden ca. 216.000 *Sensor Events* und im Spiel *Shake Me* wurden ca. 147.000 *Sensor Events* gesammelt. In der folgenden Abbildung sind einige Zeilen der Sensordaten zu sehen, die während eines *Shake Me*-Spiels vom Beschleunigungssensor aufgezeichnet wurden.

Timestamp	Acc on X	Acc on Y	Acc on Z	Accuracy
2014.09.14-20:44:52:622	-19.231932	-17.706451	18.82332	3
2014.09.14-20:44:52:632	-19.231932	-17.706451	18.82332	3
2014.09.14-20:44:52:642	-9.724928	-0.35412905	15.826844	3
2014.09.14-20:44:52:652	-16.930092	-5.3800373	18.591774	3
2014.09.14-20:44:52:662	-1.1168685	3.8954194	10.528529	3
2014.09.14-20:44:52:674	-16.930092	-5.3800373	18.591774	3

Abbildung 6.2: Abschnitt aus einer .csv-Datei

Am wenigsten wurde das Spiel *Long Distance* gespielt, wobei ca. 146.000 *Sensor Events* gesammelt wurden. Die Anzahl der gesammelten *Sensor Events* in *Long Distance* ist fast so viel wie in *Shake Me*, obwohl *Long Distance* deutlich weniger gespielt wurde als *Shake Me*. Der Grund dafür ist die Spieldauer der beiden Spiele. Die Spieldauer von *Long Distance* beträgt eine Minute, wobei die Spieldauer von *Shake Me* nur 10 Sekunden beträgt.

7 Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde *Acremode* entwickelt und veröffentlicht, um Sensordaten zu sammeln. Diese Arbeit wurde durch die Forschungsmethode *Research in the Large* umgesetzt. Die Verbreitung der mobilen Anwendung gehört zu den wichtigen Faktoren im Rahmen dieser Forschungsmethode. Die Sensordaten werden durch die Verwendung der mobilen Anwendung durch die Benutzer gesammelt. Nur so wird eine sehr hohe Anzahl an Sensordaten erreicht. Diese Sensordaten können als Trainingsdaten verwendet werden. Problematisch dabei ist die Annotation der Trainingsdaten. Es kann nicht davon ausgegangen werden, dass die Nutzer in der gegebenen Zeitspanne durchgehend die geforderte Aktivität ausführen. Deshalb ist die Verwendung der Sensordaten als Trainingsdaten bedenklich.

Um eine hohe Anzahl an Sensordaten zu erreichen, sollte die App attraktiv für die Benutzer gestaltet sein. Die Datenerhebung begann direkt nach der Veröffentlichung auf *Google Play*, als die ersten Benutzer angefangen haben, *Acremode* zu spielen. Zur Auswertung wurden alle gesammelten Daten aus dem Server heruntergeladen.

Diese Arbeit untersuchte die nutzerzentrierte Datenerhebung zur Aktivitätserkennung. Verschiedene Sensordaten wurden bei unterschiedlichen Aktivitäten mit Hilfe der eigens entwickelten mobilen Anwendung für Android gesammelt. Die gesammelten Sensordaten lassen die Ausrichtung des mobilen Endgeräts bei den Aktivitäten erkennen. Der bedeutendste Sensor für die Aktivitätserkennung von alltäglichen Tätigkeiten ist der Beschleunigungssensor [9]. Der Beschleunigungssensor ist auch der Sensor, der, im Vergleich zum Licht- oder Annäherungssensor, die meisten Sensordaten geliefert hat. Wird abschließend die Forschungsmethode *Research in the Large* und die Verwendung von der entwickelten App zur Erhebung von Sensordaten betrachtet, wird ersichtlich, dass diese Methode dazu beiträgt viele Daten zu erreichen. Wichtig ist auch die Verbreitung der Anwendung und somit die Anzahl der Benutzer. *Acremode* kann für neue Forschungsfragen eingesetzt werden, indem das Spiel erweitert wird. Somit gibt es eine bereits erreichte Benutzermenge für neue Forschungsfragen.

Literaturverzeichnis

- [1] J. W. Lockhart, T. Pulickal, and G. M. Weiss, "Applications of mobile activity recognition," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing, UbiComp '12*, (New York, NY, USA), pp. 1054–1058, ACM, 2012.
- [2] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: The design, implementation and evaluation of the cenceme application," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, SenSys '08*, (New York, NY, USA), pp. 337–350, ACM, 2008.
- [3] A. Kansiz and M. Guvensan, "Mobil telefon ile kaza tespiti accident recognition using mobile phone," in *Signal Processing and Communications Applications Conference (SIU), 2013 21st*, pp. 1–4, April 2013.
- [4] W. Wibisono, D. Arifin, B. Pratomo, T. Ahmad, and R. Ijtihadie, "Falls detection and notification system using tri-axial accelerometer and gyroscope sensors of a smartphone," in *Technologies and Applications of Artificial Intelligence (TAAI), 2013 Conference on*, pp. 382–385, Dec 2013.
- [5] B. Poppinga, H. Cramer, M. Böhmer, A. Morrison, F. Bentley, N. Henze, M. Rost, and F. Michahelles, "Research in the large 3.0: App stores, wide distribution, and big data in mobilehci research," in *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services Companion, MobileHCI '12*, (New York, NY, USA), pp. 241–244, ACM, 2012.
- [6] N. Henze and M. Pielot, "App stores: External validity for mobile hci," *interactions*, vol. 20, pp. 33–38, Mar. 2013.
- [7] C. Spinuzzi, "Using a handheld pc to collect and analyze observational data," in *Proceedings of the 21st Annual International Conference on Documentation, SIGDOC '03*, (New York, NY, USA), pp. 73–79, ACM, 2003.
- [8] S. Kaghyan and H. Sarukhanyan, "Accelerometer and gps sensor combination based system for human activity recognition," in *Computer Science and Information Technologies (CSIT), 2013*, pp. 1–9, Sept 2013.
- [9] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explor. Newsl.*, vol. 12, pp. 74–82, Mar. 2011.
- [10] Google, "Sensor types supported by the Android platform.." http://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-intro. [Online; accessed 01-October-2014].
- [11] N. Henze, A. S. Shrazi, A. Schmidt, M. Pielot, and F. Michahelles, "Empirical research through ubiquitous data collection," *Computer*, vol. 46, no. 6, pp. 74–76, 2013.

- [12] S. Schneegass, F. Steimle, A. Bulling, F. Alt, and A. Schmidt, “Smudgesafe: Geometric image transformations for smudge-resistant user authentication,” in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, (New York, NY, USA), pp. 775–786, ACM, 2014.
- [13] N. Henze, E. Rukzio, and S. Boll, “100,000,000 taps: Analysis and improvement of touch performance in the large,” in *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, (New York, NY, USA), pp. 133–142, ACM, 2011.
- [14] N. Henze, E. Rukzio, and S. Boll, “Observational and experimental investigation of typing behaviour using virtual keyboards for mobile devices,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, (New York, NY, USA), pp. 2659–2668, ACM, 2012.
- [15] M. Pielot, N. Henze, and S. Boll, “Experiments in app stores - how to ask users for their consent?,” 05 2011.
- [16] Google, “Coordinate system (relative to a device) that’s used by the Sensor API.” http://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords. [Online; accessed 01-October-2014].

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.

Ort, Datum, Unterschrift