Institut für Visualisierung und Interaktive Systeme
Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Fachstudie Nr. 160

Räumliche Kommando-basierte Interaktion mit Kinect

Verena Käfer Peter Vollmer Niklas Schnelle

Studiengang:	Softwaretechnik
Prüfer:	Prof. Dr. Albrecht Schmidt
Betreuer:	Dipl.-Inf. Bastian Pfleging
begonnen am:	18. Juni 2012
beendet am:	18. Dezember 2012
CR-Klassifikation:	H.4.1, K.1

Kurzfassung

Im Rahmen dieser Fachstudie beschäftigen wir uns mit der Konzeption und Implementierung eines Systems zur Auslösung Computer-basierter Aktionen durch Interaktion mit dem Raum in Form von Körperpositionen. Hierzu besprechen wir insbesondere die Entwicklung eines erweiterbaren Prototypen mit Kinect[1]-basierter Erkennung der Körperhaltung und durch den Benutzer konfigurierbarer Aktions und Auslöserwahl.
Zur Evaluierung des Protoypen wurde zudem eine Benutzerstudie durchgeführt in der wir sowohl den Usability Index SUS[2] zur Feststellung der Benutzerfreundlichkeit bestimmen sowie die Frage klären wie die größe der auslösenden Raumbereiche sowie das kurzzeitige Erinnern an ihre Position das Auslöseverhalten verändern.

[1] http://www.microsoft.com/en-us/kinectforwindows/
[2] http://www.measuringusability.com/sus.php

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1. Einleitung

1.1. Motivation

Eines der heißesten Themen bei der Frage, wie sich unser Leben und unsere Arbeit mit Computern verändert, ist die Vorstellung des Ubiquitären Computings, bei dem Computer mit den verschiedensten Alltagsgegenständen verschmelzen. Sie werden zu eingebetteten, nicht mehr getrennt wahrnehmbaren Bestandteilen unserer Umgebung, von Lampen mit Farbwahl über Waschmaschinen, die automatisch bei günstigem Strom waschen, bis hin zur Steuerung der gesamten Hauselektronik.
Wie bei jeder Entwicklung ergeben sich hierdurch vielerlei neue Herausforderung, wie die notwendige aber schwierige Standardisierung der Schnittstellen oder die enormen Anforderungen an die Zuverlässigkeit solch allgegenwärtiger Systeme. Zu diesen Herausforderungen des Ubiquitären Computings gehört auch die Frage, wie der Mensch mit einer solchen Vielzahl der Systeme komfortabel und möglichst natürlich umgehen kann, ohne dabei die Vielseitigkeit der Einsatzmöglichkeiten zu gefährden.
Da die eingebetteten Systeme in der gesamten Umgebung, sowohl im privaten als auch im öffentlichen Umfeld, verteilt sind, fallen zentrale Steuerungspunkte wie die Bedienung über einen PC oder Fernseher als mögliche Interaktionsform in vielen Fällen aus. Es scheint daher sinnvoll, die Interaktion hin zum gesteuerten Gerät zu verlagern, was auch der Natürlichkeit der Information zu Gute kommen kann, da der Bezug zum bedienten Gerät so leichter herzustellen ist.
Bei weiträumig verteilten Ubiquitären Systemen scheint es daher sinnvoll, auch die interaktive Schnittstelle zum System im Raum zu verteilen. Genau hier setzt unser System an und ermöglicht es dem Nutzer den Raum selbst durch seine Körperbewegung und Gesten zur Mensch-Computer-Schnittstelle werden zu lassen. Weiterhin erscheint es sinnvoll, dass nicht jedes Gerät die Hardware zur Interaktion selbst mitbringen muss, gerade bei sehr kleinen tief eingebetteten Systemen würde dies nicht nur die Kosten steigen lassen, sondern auch die Komplexität der einzelnen Komponenten erhöhen, was mit den bekannten Problemen erschwerter Wartbarkeit, schwierigerer Standardisierung et cetera verbunden ist.

1.2. Verwandte Arbeiten

Spätestens seit der Markteinführung der Kinect für Microsofts XBox 360 im Jahr 2010 ist die Interaktion mit Computern (im weitesten Sinne) durch Gesten und Posen im Raum Teil des Alltags geworden. Jedoch gab es auch schon vorher Bestrebungen, Menschen und

Computer auf natürliche Weise im Raum interagieren zu lassen. So wurde zum Beispiel schon 2005 Fußboden als räumliches Eingabemedium genutzt [PKLLO05]. Mit Sonys EyeToy Kamera und Microsofts Kinect schließlich zog Mensch-Computer-Interaktion mit Hilfe von Körperposen und Gesten ins Wohnzimmer ein.
Es liegt daher nah, diese spielerische Form der Interaktion auch zum Arbeiten, in der Visualisierungstechnik und in besonderen Situationen wie im Operationssaal [GPC11] oder in der Küche [Pan12] einzusetzen.

1.3. Abstrakter Aufbau des Systems

Um die Nützlichkeit und Bedienungsfreundlichkeit des Konzepts einer verteilten interaktiven Schnittstelle zu untersuchen, haben wir mit der Konzeption und Entwicklung eines Prototypensystems begonnen. Dabei haben wir uns als Sensorkomponente für die Detektion und Erkennung der menschlichen Körperhaltung im dreidimensionalen Raum für die Nutzung von Microsofts Kinect[1] Tiefenbildkamera entschieden. Sie bietet nicht nur eine hohe Erkennungsqualität für die menschliche Körperhaltung, sowie einen relativ großen Abdeckungsbereich, sondern ist im Vergleich zu 3D-Motion-Trackingsystemen auch mit einem sehr ausgereiften und leicht zu nutzenden SDK (software development kit) programmierbar.
Als weitere Kernkomponente des Systems wählten wir einen Handelsüblichen PC auf dem das System konfiguriert wird und bei dem davon ausgegangen werden kann, dass mit relativ wenig Aufwand eine Steuerung eines jeden verteilten Systems über Netzwerkverbindungen oder speziell hierfür konzipierter Bussysteme realisierbar ist.
Aus diesen Systemkomponenten ergeben sich dabei natürlich auch Einschränkungen in der Funktionalität des entwickelten Prototypen. Die Reichweite der Kinect beläuft sich zum Beispiel zwar auf beachtliche 4 m, jedoch ist nur der Bereich in der Blickrichtung des Sensors abdeckbar, hinzu kommt, dass die Erkennung der menschlichen Körperposition zumeist auf der Kinect zugewandte Orientierung ausgelegt ist.

[1] http://www.microsoft.com/en-us/kinectforwindows/

2. Konzept

2.1. Aufbau des Programms

Mit unserem Programm ist es möglich, Actions und Trigger zu definieren und diese dann zu verbinden.

- **Actions** sind hierbei die Aktionen, die ausgelöst werden sollen. Das ist im jetzigen Fall zum Beispiel das Starten eines Programms aber man kann sich auch vorstellen, dass in der Zukunft auch andere Komponenten wie Licht, Fenster oder Elektrogeräte gesteuert werden können. Immer wenn ein Trigger aktiviert wird werden alle zugehörigen Actions ausgeführt.
- **Trigger** sind die Auslöser einer Action. Hierbei handelt es sich um dreidimensionale Bereiche (beispielsweise ein Würfel), deren Positionen genau bestimmt werden können. Dazu können aus einer Liste die auslösenden Körperteile gewählt werden (beispielsweise die rechte Schulter). Sobald sich eines dieser Körperteile im definierten Bereich befindet, werden die zugehörigen Actions ausgelöst.

Als Szenario könnte man sich vorstellen, dass der Lichtschalter am anderen Ende des Zimmers ist. Anstatt immer aufstehen zu müssen, definiert man sich einfach einen Bereich an der Schreibtischecke und verbindet ihn mit der „Licht-Action “. Sobald die Hand auf der Ecke liegt, geht das Licht an oder aus.

2.2. Erstellung von Mockups

Um eine ungefähre Vorstellung zu bekommen, wie ActOnIt aussehen soll, war unser erster Schritt Mockups zu erstellen. Nachfolgend sind die Ergebnisse zu sehen.

2.2.1. View

In dieser View (siehe Abbildung 2.1) sollte man links die Repräsentation eines Menschen und die aktiven Trigger sehen können. Rechts könnte man den Triggern über eine Combobox Actions zuordnen. Um Trigger zu aktivieren und zu deaktivieren gäbe es eine Checkbox.

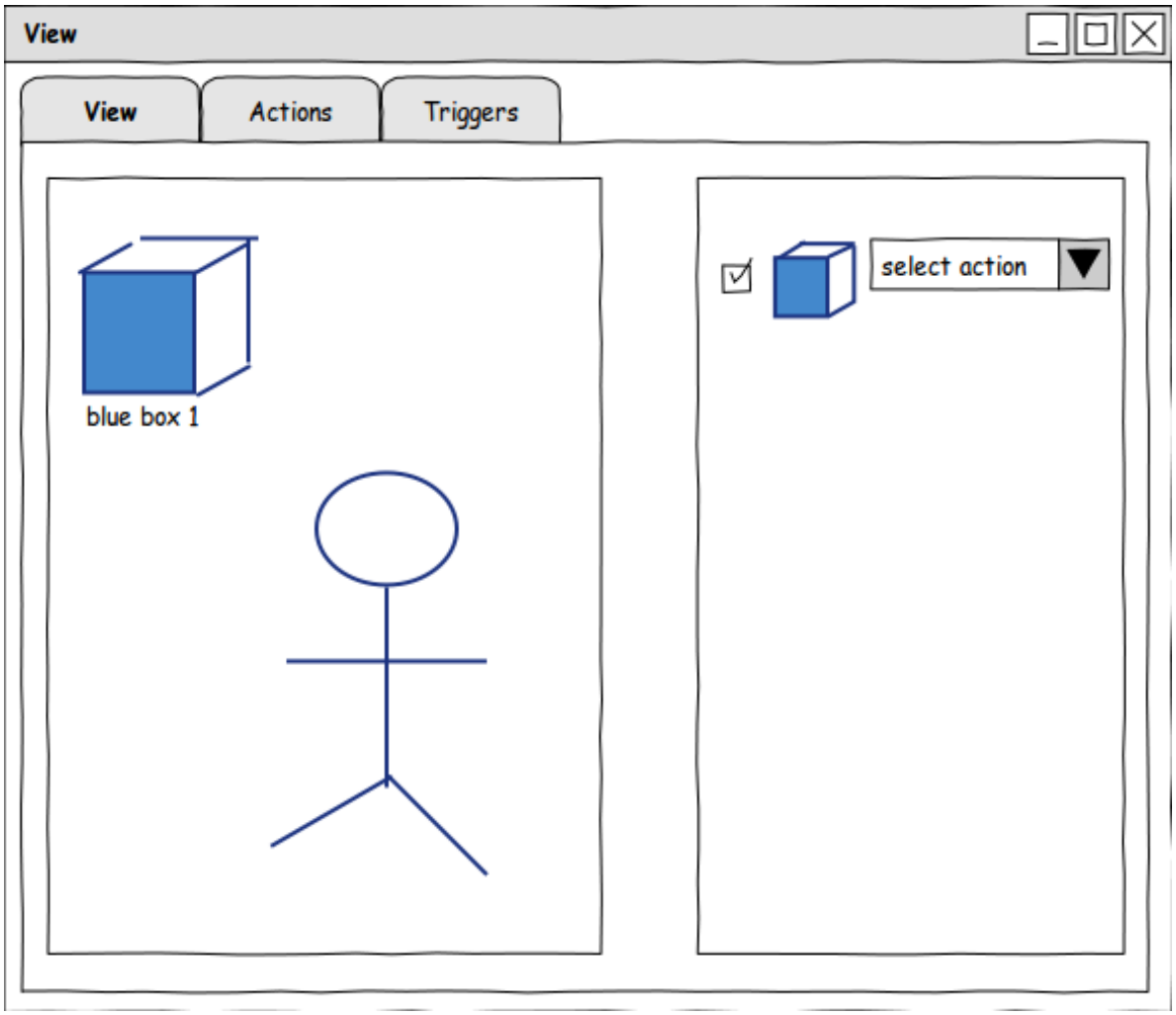

Abbildung 2.1.: Geplantes Aussehen des View-Tabs im Mockup

2.2.2. Actions

Für die Definition von Actions war vorgesehen, dass es rechts eine Liste gibt (siehe Abbildung 2.2) in der alle vorhandenen Actions aufgelistet werden, über Buttons neue Actions hinzugefügt oder gelöscht werden können. Zum Bearbeiten würde man die entsprechenden Action auswählen und links würde die zugehörige Bearbeitungsmaske erscheinen. Dort könnte man die Action bearbeiten und mit dem „Save "Button speichern.
Für unsere Fachstudie reicht es, dass für ActOnIt ein Typ von Actions bereit steht. Wir wollten, dass man Programme starten kann. Deshalb sollte im Definitionsbereich einer Action der Name geändert werden können und die Programme, die gestartet werden sollen, einstellbar sein.

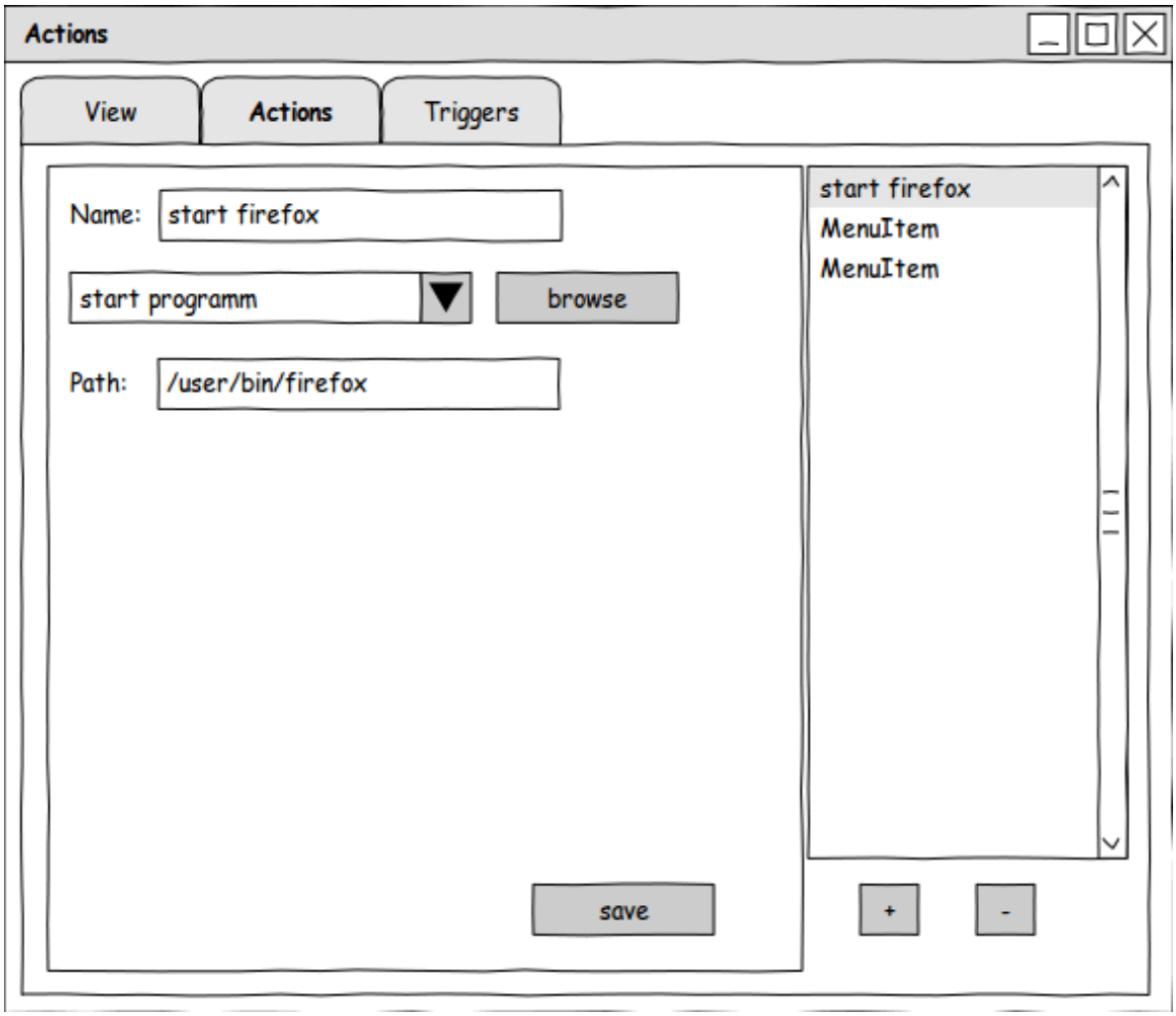

Abbildung 2.2.: Geplantes Aussehen des Action-Tabs im Mockup

2.2.3. Trigger

Für die Definition der Trigger war vorgesehen, dass es rechts eine Liste gibt (siehe Abbildung 2.3), in der alle vorhandenen Trigger angezeigt werden, über Buttons können neue Trigger hinzugefügt und ausgewählte Trigger gelöscht werden. Zum Bearbeiten würde man den entsprechenden Trigger auswählen, so dass links die zugehörige Bearbeitungsmaske erscheint. Dort könnte der Trigger bearbeitet werden, abgeschlossen wird die Bearbeitung durch einen weiteren „Save "Button. Für unsere Fachstudie reicht es, dass für ActOnIt ein Art Trigger vorhanden ist. Der Einfachheit halber entschieden wir uns für die Entwicklung eines Triggers, der auslöst sobald ein definiertes Körperteil in einem vordefinierten Würfelförmigen Bereich erkannt wird. Deshalb sollten im Definitionsbereich eines Triggers 3 Punkte definiert

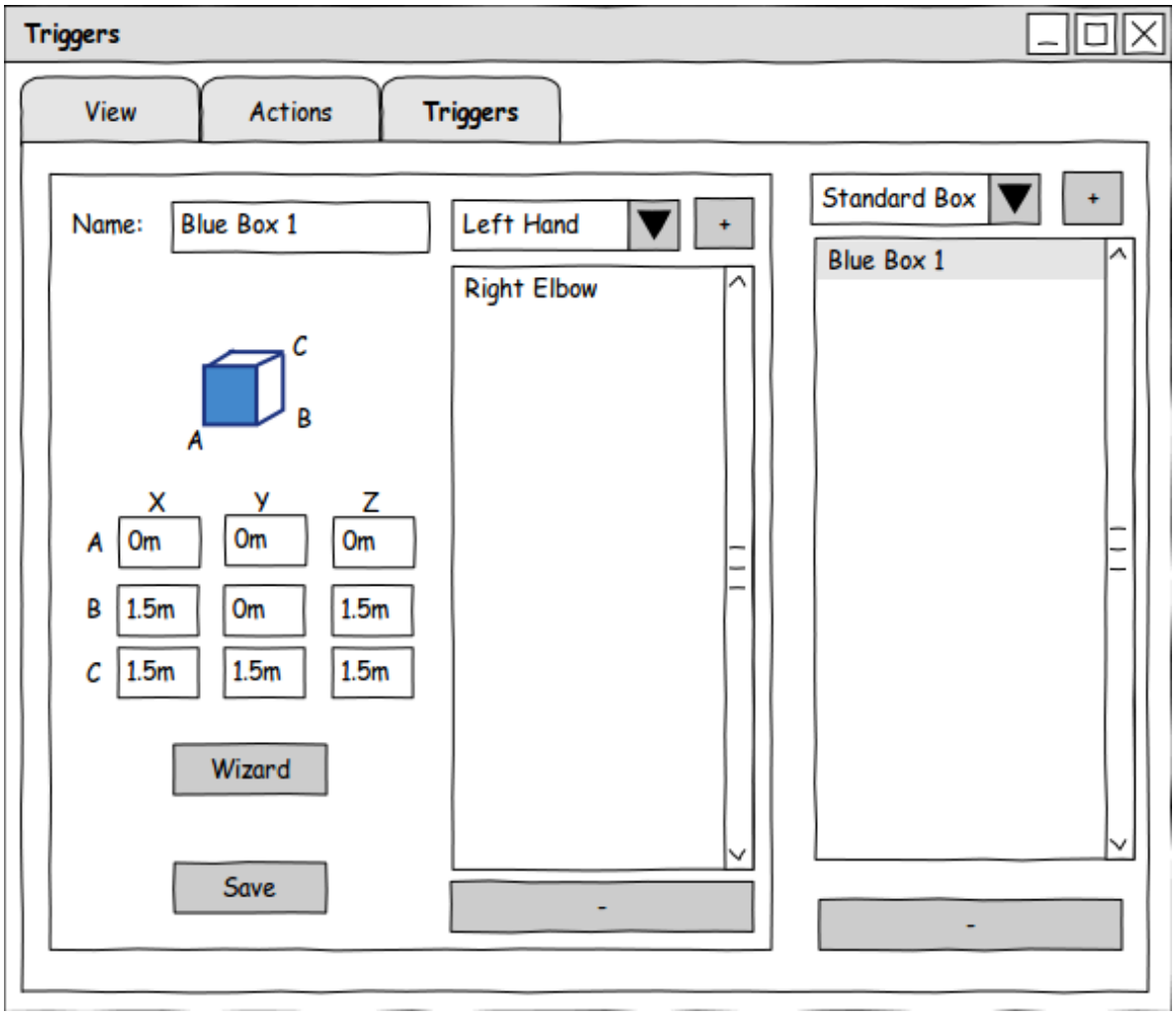

Abbildung 2.3.: Geplantes Aussehen des Trigger-Tabs im Mockup

werden können, welche die Ecken des aktiven Bereichs definieren. Außerdem können Trigger benannt werden.

2.2.4. Wizard

Für den Wizard haben wir uns ein zusätzliches Fenster überlegt, dass über einen Button in der Bearbeitungsmaske des Triggers gestartet wird (siehe Abbildung 2.4). Hier könnte man dann die relevanten Daten des Triggers, wie Länge, Breite, Höhe eingeben sowie ein Körperteil wählen, dass zum bestimmen der Triggerposition dient. Nach dem Klicken des Startbuttons beginnt der Countdown zu laufen an dessen Ende die Position des zur Einstellung gewählten Körperteils als Position des Triggers übernommen wird.

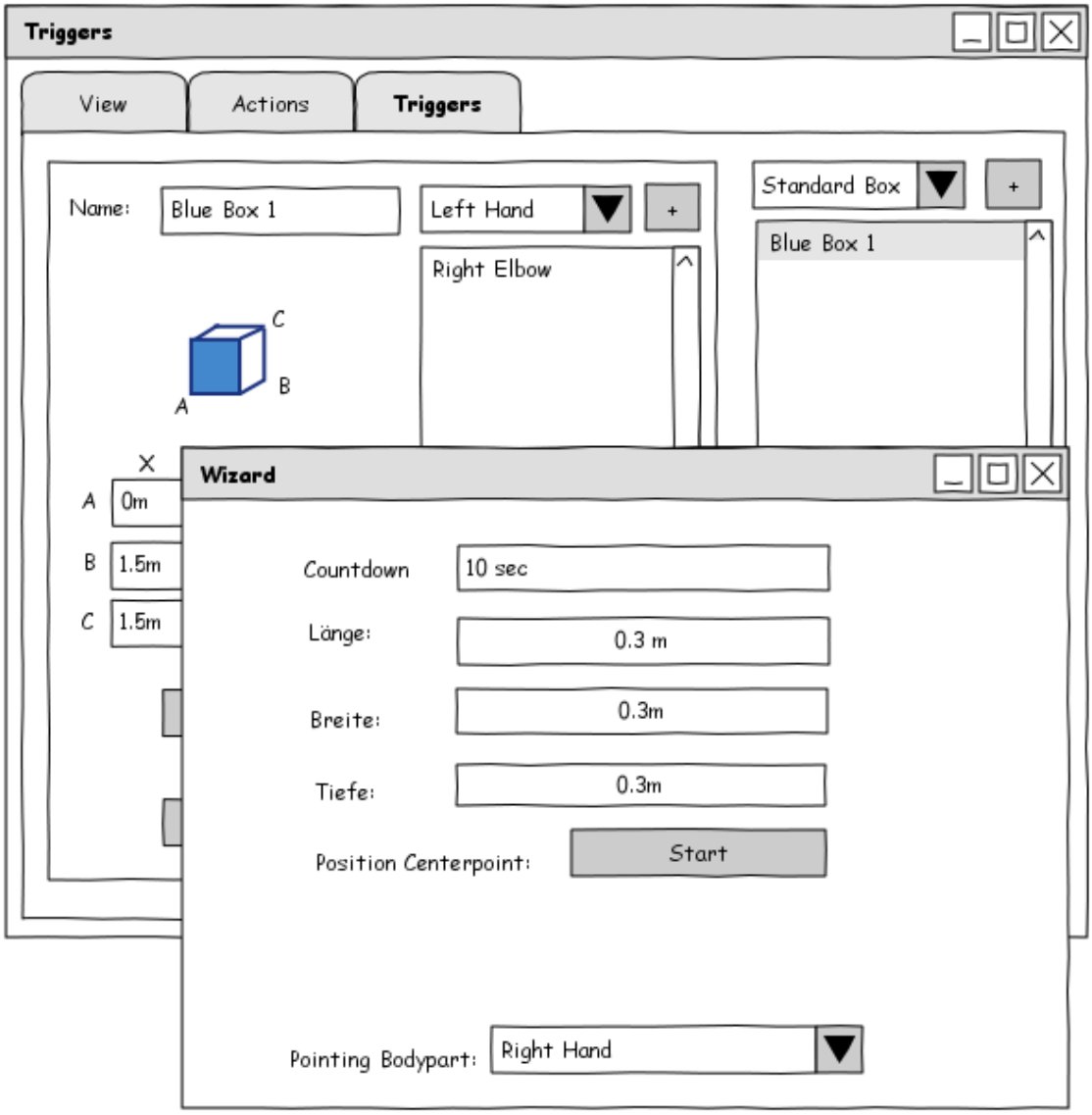

Abbildung 2.4.: Geplantes Aussehen des Wizards im Mockup

3. Implementierung

3.1. Architektur

Im Folgenden wir die Architektur von ActOnIt in einer Übersicht dargestellt.

Grundlage der Architektur von ActOnIt ist ein Client-Server-System mit eingebauter Netzwerktransparenz (siehe Abbildung 3.1) welches einen Skelett-Tracker, in unserem Fall eine Kinect, mit der Benutzeroberfläche von ActOnIt verbindet.
Dabei nutzen wir die neuste Inkarnation des sogenannten EI-Toolkits, welche im Rahmen einer anderen Fachstudie entwickelt wurde. Hierbei handelt es sich um einen Nachrichtenbasierten virtuellen Bus zur Kommunikation verschiedenster Eingabegeräte, Sensoren und Softwaresystemen, die ihre Funktionalität nutzen. Das EI-Toolkit kann dabei mit unterschiedlichen Transportmechanismen wie zum Beispiel JSON auf UDP Broadcast genutzt werden.
Da noch kein Adapter, im EI-Toolkit Stub genannt, für die Skelett-Tracking-Funktionalität der Kinect vorhanden war, haben wir diesen im Rahmen unserer Fachstudie ebenfalls entwickelt, so dass er nun auch für weitere Projekte zur Verfügung steht.

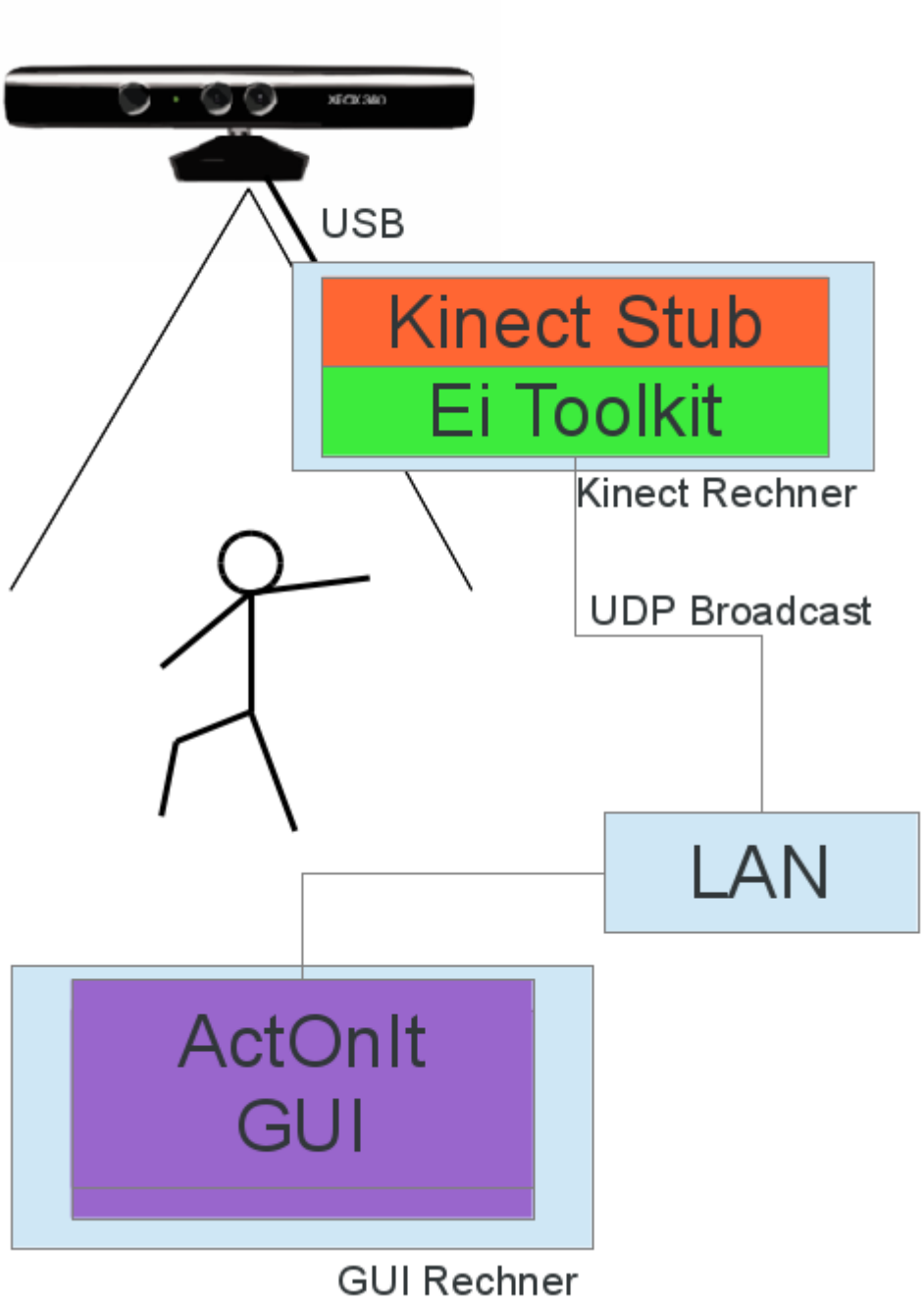

Abbildung 3.1.: Der Schematische Aufbau von ActOnIt

3.2. ActOnIt GUI

3.2.1. Unterteilung der Oberfläche

Bei der Implementierung von ActOnIt ersetzen die Connections und der immer sichtbare Viewbereich die View aus den Mockups, aus 2.2.1.

Connections

Auf der Connections-Seite kann einem Trigger eine Action zugeordnet werden, indem in der Combobox auf der rechten Seite, hinter dem entsprechenden Trigger, eine Action ausgewählt wird (siehe Abbildung 3.2). Zudem generiert ActOnIt eine Farbe, die in der dritten Spalte angezeigt wird und zur Identifizierung der farbigen Trigger im Viewbereich dient. Auf der Connections-Seite werden im Viewbereich alle Trigger und der Avatar angezeigt.

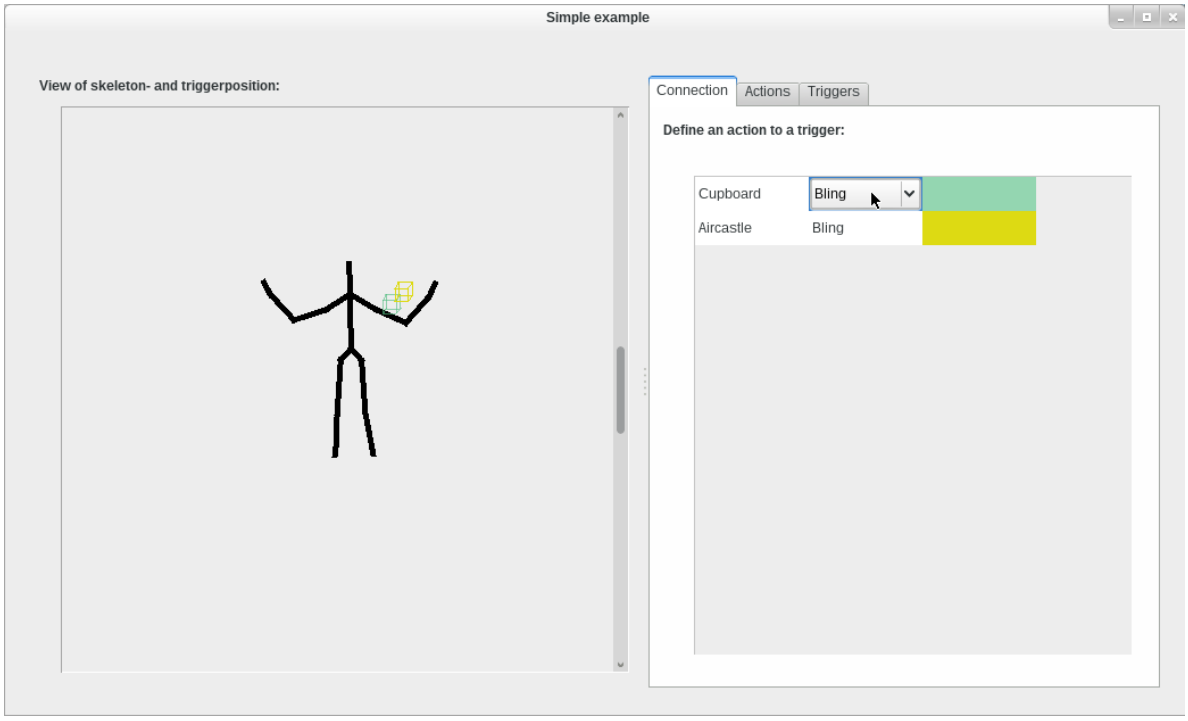

Abbildung 3.2.: Sceenshot des Connectionstab. Hier können Actions mit Triggern verbunden werden

Actions

Auf der Actions-Seite werden rechts alle existierenden Actions aufgelistet, außerdem kann mit der Combobox über der Liste der Typ der zu erstellenden Action eingestellt werden (siehe Abbildung 3.3). Hinzugefügt wird die Action dann über den Button „+ “. Ausgewählte Actions können über „- “unterhalb der Liste entfernt werden. Der mittlere Bereich ändert sich je nach ausgewählter Action und deren Typ. Darin wird die ausgewählte Action definiert. Beim Typ „Open Program Action “kann der Name der Action und der Pfad zum auszuführenden Programm geändert werden. Im Gegensatz zum unserem Mockup der Actions, aus 2.2.2, muss man nicht speichern, da jede Änderung automatisch gespeichert wird. Auf dieser Seite werden im Viewbereich alle definierten Trigger und der Avatar angezeigt.

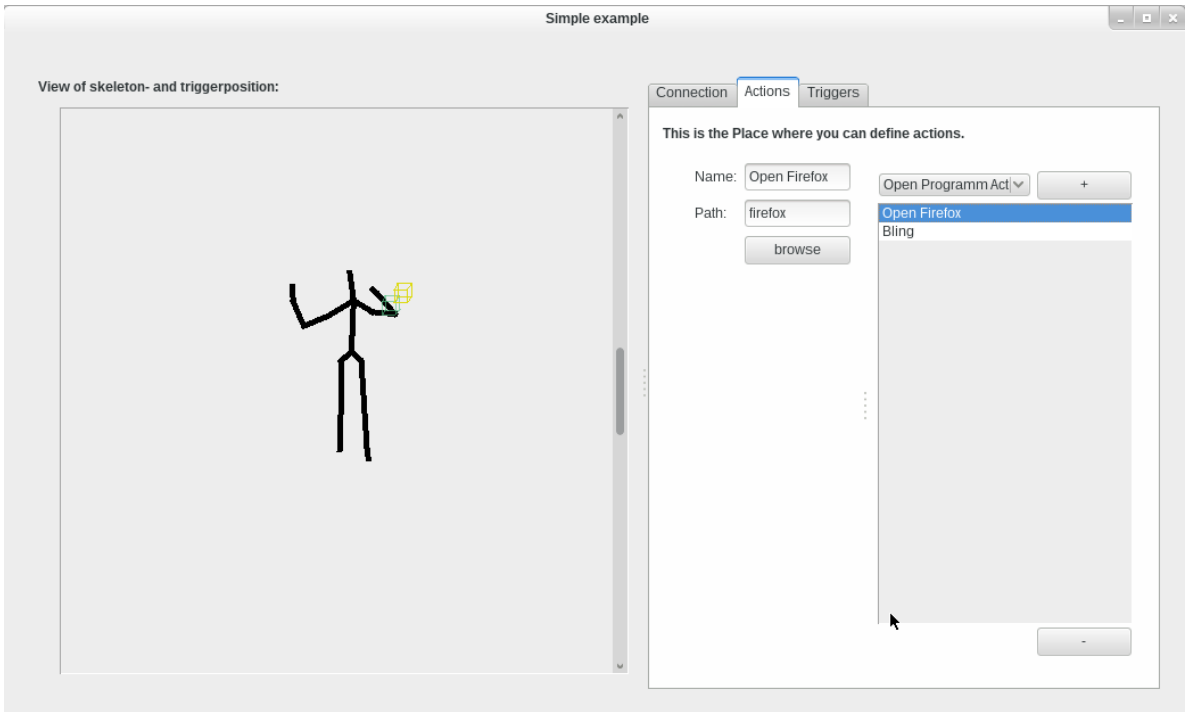

Abbildung 3.3.: Sceenshot des Actionstab. Hier werden die Actions erstellt und verwaltet

Trigger

Auf der Trigger-Seite werden auf der rechten Seite alle existierenden Trigger aufgelistet, außerdem kann mit der Combobox über der Liste der Typ des zu erstellenden Triggers eingestellt werden (siehe Abbildung 3.4). Hinzugefügt wir die Action dann über den Button „+ “. Ausgewählte Actions können über den Minusbutton unterhalb der Liste entfernt werden. Der mittlere Bereich ändert sich je nach ausgewähltem Trigger und dessen Typ. Darin wird dann der ausgewählte Trigger definiert. Beim Typ Boxtrigger kann der Namen des Trigger geändert werden. Außerdem existiert eine Liste, die die Körperteile auflistet, die den Trigger auslösen können. Der Typ des Körperteils wird mit der Combobox über der Liste ausgewählt und mit dem Button „+ “daneben hinzugefügt. Mit dem Button „- “unter dieser List kann ein ausgewähltes Körperteil entfernt werden. Im Gegensatz zu unserem Mockup der Trigger , aus 2.2.3, muss man nicht speichern, da jede Änderung automatisch gespeichert wird. Auch kann die Größe und Position des Boxtriggers nur noch über den Wizard einsgestellt werden. Der Wizard, in 3.2.1, wird über den Button „Wizard “unterhalb der Liste gestartet. Auf dieser Seite werden im Viewbereich nur der ausgewählte Trigger und der Avatar angezeigt.

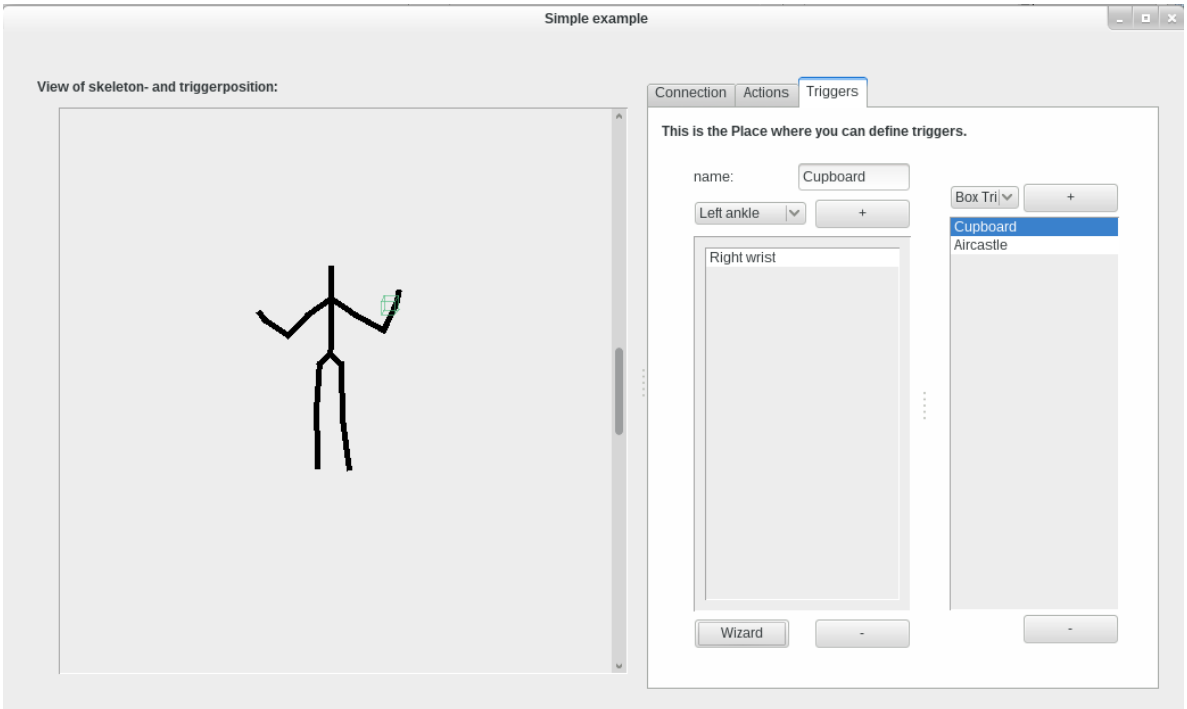

Abbildung 3.4.: Sceenshot des Triggertabs. Hier werden die Trigger erstellt und verwaltet

Wizard

Mit diesem Wizard wird die Größe und Position des ausgewählten Triggers definiert (siehe Abbildung 3.5). Hier für den Typ „Boxtrigger “. Dazu stellt man mit der Combobox das Körperteil ein, mit dem man die Position des Triggers bestimmen möchte. Anschließend stellt man die Größe der Box ein und klickt den Button „start countdown “. Danach hat man 5 Sekunden Zeit das eingestellte Körperteil an die richtige Stelle zu setzen. Wenn der Countdown abgelaufen ist wird diese Position benutzt um eine Box zu errechnen. Klickt man dann noch auf den Button „Finish “, dann wird die Box gespeichert. Ansonsten kehrt man zu dem Zustand vor dem Wizard zurück.

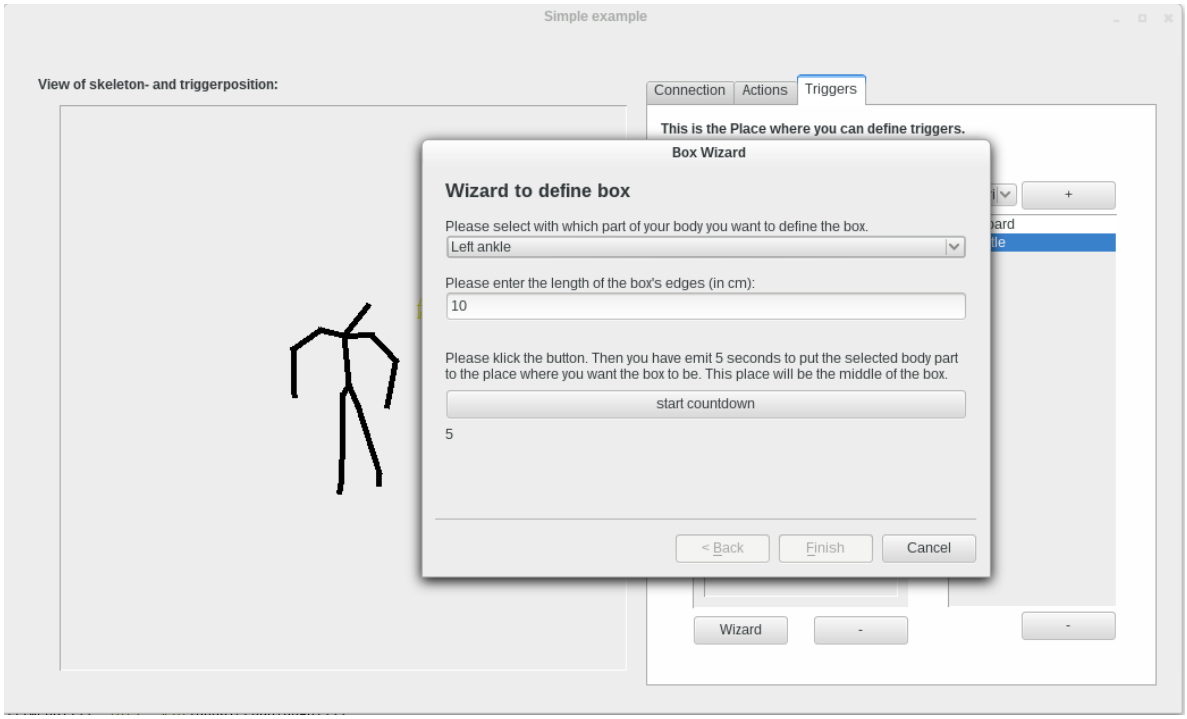

Abbildung 3.5.: Sceenshot des Wizards. Er hilft die Position eines Triggers festzulegen

3.2.2. Koppelung von Actions und Triggern

Intern werden Actions und Trigger über das Signal/Slot-System von Qt verbunden, dies ermöglicht es das Aufrufen von Actions durch Trigger mit minimalem Codeaufwand zu erreichen und bietet dennoch genug Flexibilität. So könnte das System leicht so erweitert werden, dass ein Trigger mehrere Actions auslöst und auch die Anbindung neuer Trigger- und Actiontypen ist leicht möglich. Ein weiterter Vorteil der sich durch das Aufsetzen auf Signals/Slots ergibt, ist die Möglichkeit Qt's Support für das Aufrufen eines Slots in einem anderen Thread über einen speziellen Warteschlangenmechanismus nutzen zu können, so das Actions auch in anderen Threads ausgelöst werden könnten.

3.2.3. Laden und Speichern

Alle Actions und Trigger werden in einer XML-Datei gespeichert. Diese befindet sich im config-Verzeichnis des PCs. Der XML-Baum sieht beispielhaft folgendermaßen aus:

```
<actonit>
    <actions>
        <OpenProgramAction file_name="firefox" description="opens a program"
        id="{ca164b67-5db0-40e4-8bc3-786099499eae}" name="a"/>
    </actions>
    <triggers>
        <BoxTrigger description="Standard box defined with 3 corners"
        action="{ca164b67-5db0-40e4-8bc3-786099499eae}" name="b">
            <coordinates>
                <A x="0" y="0" z="0"/>
                <B x="0" y="0" z="0"/>
                <C x="0" y="0" z="0"/>
            </coordinates>
            <bodyParts>
                <part part="0"/>
            </bodyParts>
        </BoxTrigger>
    </triggers>
</actonit>
```

Nach dem Schließen von ActOnIt wird zuerst die Liste der Actions in die XML-Datei geschrieben. Darauf folgt die Liste der Trigger. Jeder Trigger hat eine zugehörige Action gespeichert, die durch eine eindeutige ID identifizierbar ist.
Beim Öffnen von ActOnIt werden zuerst alle Actions geladen. Dabei wird jede Action mit Ihrer zugehörigen ID in eine Hashmap gespeichert. Danach werden die Trigger geladen. Nun

wird die ID in der zugehörigen Hashmap nachgeschlagen und die entsprechende Action mit dem Trigger verknüpft.

3.2.4. Anbindung an Kinect und Geräteabstraktion

Aufgrund der Architektur von ActOnIt bei der die Benutzeroberfläche über das EI-Toolkit Skelett-Tracking-Daten bezieht erreichen wir eine inhärente Geräteunabhängigkeit. Die Benutzeroberfläche erwartet zwar eine relativ anwendungsspezifische Skelett-Repräsentation, jedoch könnte diese mit den Daten eines beliebigen Sensors berechnet werden. So wäre es zum Beispiel möglich, eine kompatible Anbindung eines Mehrkamera-Tracking-Systems wie sie beim Motion Capture verwendet werden, zu benutzen oder auch eine Tiefenbildkamera eines anderen Herstellers. Notwendig hierfür ist allerdings, zumindest wenn bereits definierte Trigger beibehalten werden sollen und keine Anpassungen der Konstanten zur Darstellung vorgenommen werden, dass die alternativen Systeme in das gleiche Koordinatensystem übertragen werden.

4. Nutzerstudie

4.1. Ziel

Mit dieser Studie verfolgen wir zwei Ziele. Zum einen wollen wir heraus finden, ob kleinere Trigger schlechter getroffen werden als große. Zum anderen wollen wir evaluieren, ob und wie viel länger die Probanden brauchen, um einen Trigger zu treffen, wenn sie sich zwischenzeitlich mit anderen Triggern beschäftigt waren. Zusätzlich wollen wir über eine Befragung herausfinden, wie das System angenommen wird.

4.2. Aufbau

Die Kinect steht mittig unter einem großen Bildschirm (128cm x 75cm) an der Wand in ca 100 cm Höhe. Der Proband steht zwei Meter von der Kinect entfernt. Rechts von dem Probanden steht ein Stehtisch mit einer Maus und Tastatur darauf. Die Position des Probanden wird durch ein $1m^2$ Rechteck auf dem Boden begrenzt. Mit dem Bildschirm ist ein Rechner mit Actonit und einem benötigten Soundprogramm verbunden. Über das Netzwerk ist der Rechner außerdem mit dem Kinect-Server verbunden. Der Proband kann mit der Maus und der Tastatur Actonit bedienen.

4.3. Durchführung

Die Studie gliedert sich in drei Phasen:

4.3.1. Phase 1

Dem Probanden wird Actonit und das Action-Trigger-Prinzip erklärt. Daraufhin soll er drei gleich große Trigger (10 cm) sowie Actions erstellen und verbinden. Dafür werden ausführbare Scripte bereitstehen, die per Ton „eins “, „zwei“und „drei“ausgeben. Nun soll der Proband die drei Trigger so schnell wie möglich in aufsteigender Reihenfolge auslösen. Dabei wird die Zeit gemessen.

4.3.2. Phase 2

Die Probanden legen die Position von vier bereits existierenden Triggern „oben“, „unten“, „rechts“und „links “fest. Die zugehörigen Actions steuern ein Musikprogramm. Dabei enspricht „oben “„lauter “, „unten“„leiser“, „rechts“„nächstes Lied“und „links“„vorheriges Lied“. Daraufhin sollen sie kleinere Aufgaben (siehe unten) durchführen. Davon wird jeweils die Zeit gemessen. Das Ganze wird drei mal mit den Triggergrößen 10 cm, 20 cm und 40 cm durchgeführt. Die Reihenfolge der Größen alterniert von Proband zu Proband.

Aufgabenstellungen für die Probanden

- Zwei Lieder nach vorne springen.
- Ein Lied zurück springen.
- Einmal lauter machen.
- Zweimal leise machen.
- Erst ein Lied nach vorne und dann zweimal lauter.
- Einmal leiser und danach zwei Lieder zurück

4.3.3. Phase 3

Die Probanden sollen die Trigger aus Phase 1 so schnell wie möglich in aufsteigender Reihenfolge nacheinander aktivieren. Dabei wird die Zeit gemessen.

4.4. Unabhängige Variablen

Zu den unabhängigen Variablen gehört zunächst die Kinect, da es sich immer um die gleiche Kamera handelt. Dazu kommen sämtliche anderen verwendeten technischen Geräte. In Phase 1 sind die auslösbaren Actions sowie die Triggergröße unabhängig.
In Phase 2 sind die Triggergrößen und die Aufgaben unabhängig.
In Phase 3 sind die Trigger aus Phase 1 unabhängig.

4.5. Abhängige Variablen

In Phase 1 sind die Triggerposition sowie die Zeit, die man zum Auslösen braucht, abhängige Variablen.
In Phase 2 sind die Positionen der Trigger sowie die Zeit, um die Aufgaben zu erfüllen, abhängig.
In Phase 3 ist nur die Zeit, um alle Trigger auszulösen, abhängig.

4.6. Hypothesen

Unsere Hypotese in Phase 1 und Phase 3 ist, dass eine zwischenzeitliche Unterbrechung Auswirkungen auf die zum Finden der Trigger benötigte Zeit sowie die Fehleranzahl hat. WIr vermuten, dass sich die Zeit verlängert.
In Phase 2 haben wir die Vermutung, dass die Größe der Trigger starken Einfluss auf die zum Finden der Trigger benötigte Zeit sowie die Fehleranzahl hat. Wir vermuten, dass die Zeit mit der Größe abnimmt uns sich die Fehlerzahl vergrößert.

4.7. Ergebnis

Alle Zeiten wurden in Sekunden (s) gemessen.

4.7.1. Phase 1 und Phase 3

Gruppen	Anzahl	Summe	Mittelwert	Varianz
Phase 1	18	438	24,33	302,47
Phase 3	18	396	22	171,16

Tabelle 4.1.: Zusammenfassung der gemessenen Zeiten in Phase 1 und Phase 3 in Sekunden

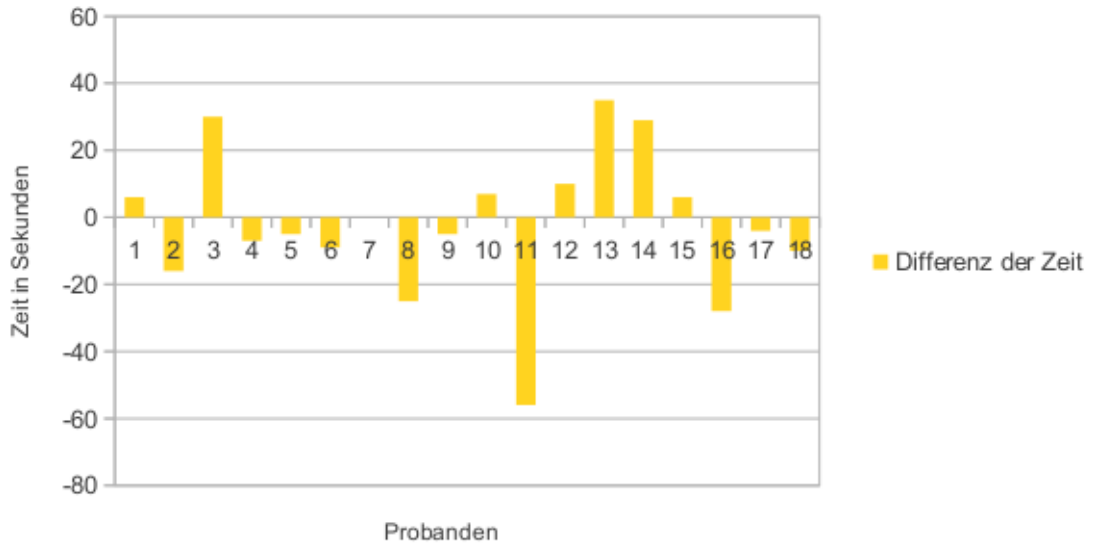

Abbildung 4.1.: Die Differenzen der Zeiten zwischen der ersten und der dritten Phase

Tabelle 4.1 zeigt die statistischen Werte für die gemessene Zeiten in Phase 1 und Phase 3.
Die zwischenzeitlich andere Beschäftigung beeinflusst die Zeit nicht statistisch signifikant, $F = 0.21$, $p = 0.65$. Offensichtlich spielt es keine Rolle, ob man sich dazwischen mit anderen Triggern beschäftigt. Bei der einen Hälfte der Probanden wurden die Zeiten schlechter, bei der anderen Hälfte besser.
In Abbildung 4.1 sieht man, dass sich keine einheitliche Struktur feststellen lässt.

Gruppen	Anzahl	Summe	Mittelwert	Varianz
Phase 1	18	44	2,44	5,08
Phase 3	18	17	0,94	1,23

Tabelle 4.2.: Zusammenfassung der Fehlerwerte in Phase 1 und Phase 3

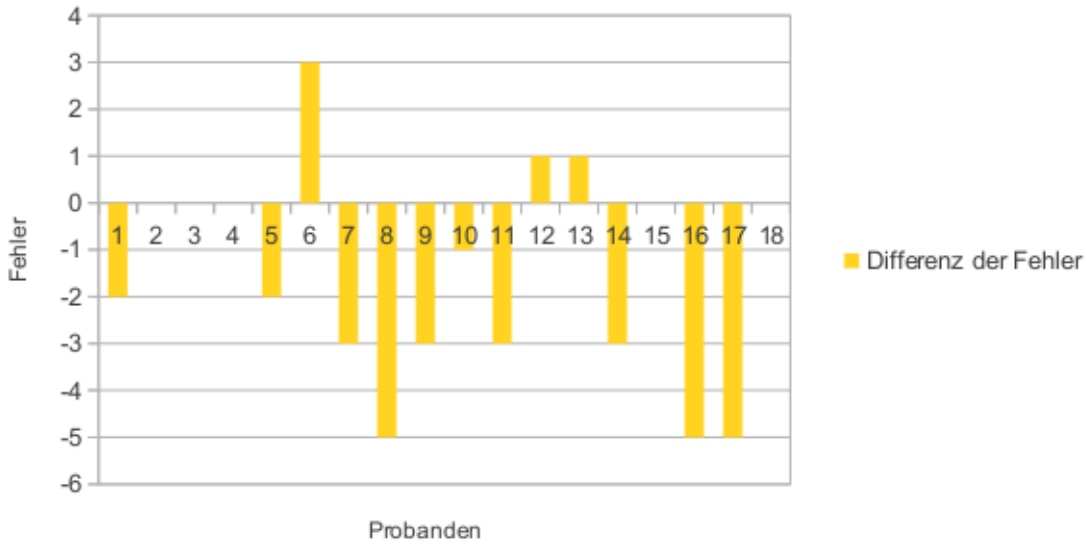

Abbildung 4.2.: Die Differenzen der Fehler zwischen der ersten und der dritten Phase

In Abbildung 4.2 sieht man deutlich, dass die meisten Probanden in Phase 3 weniger Fehler als in Phase 1 gemacht haben.

Tabelle 4.2 zeigt die Fehlerwerte in Phase 1 und Phase 3.
Die zwischenzeitlich andere Beschäftigung wirkt sich statistisch signifikant auf die Fehlerzahl aus, $F = 6.41$, $p = 0.02$.
Die Gesamtfehlerzahl ist von 44 auf 17 gesunken.

4.7.2. Phase 2

Gruppen	Anzahl	Summe	Mittelwert	Varianz
10 cm	18	1747	97,06	4206,88
20 cm	18	730	40.56	849,67
40 cm	18	452	25,11	353,63

Tabelle 4.3.: Zusammenfassung der gemessenen Zeiten in Phase 2 in Sekunden

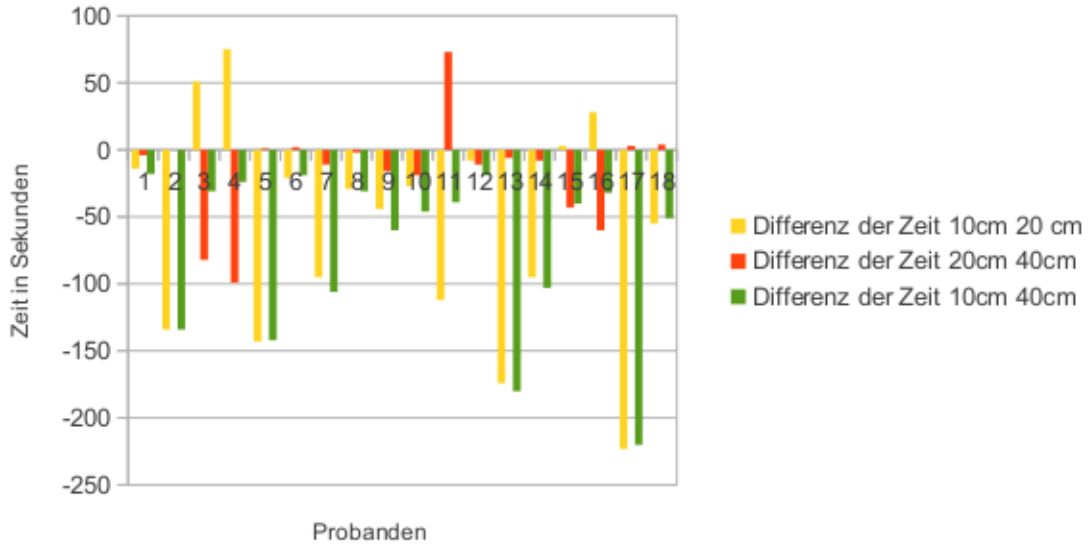

Abbildung 4.3.: Die Differenzen der Zeiten zwischen den verschiedenen Triggergrößen in Phase 2

Tabelle 4.3 zeigt die gemessenen Zeitwerte aus Phase 2.
Die Größe der Trigger wirkt sich statistisch signifikant auf die Fehlerzahl aus, $F = 14.32$, $p = 0.000012$.
Hierbei sank die Durchschnittszeit von 97,06s (10 cm) auf 40,56s (20 cm) und dann auf 25,11s (40 cm).
In Abbildung 4.3 sieht man, dass die Zeiten zum nächst größeren Trigger abgenommen haben.

Gruppen	Anzahl	Summe	Mittelwert	Varianz
10 cm	18	46	2,56	4,03
20 cm	18	56	3,11	11,40
40 cm	18	84	4,67	7,76

Tabelle 4.4.: Zusammenfassung Fehler Phase 2

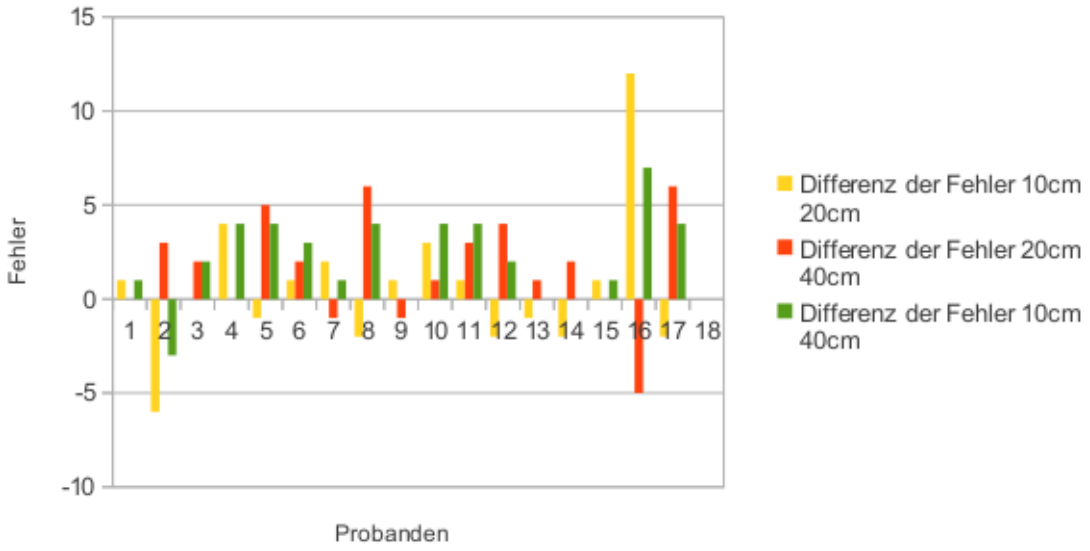

Abbildung 4.4.: Die Differenzen der Fehler zwischen den verschiedenen Triggergrößen in Phase 2

Tabelle 4.4 zeigt die Fehlerwerte aus Phase 2. Die Triggergröße wirkt sich auch statistisch signifikant auf die Fehlerzahl aus, $F = 2.79$, $p = 0.07$.
Hierbei stieg die Gesamtfehlerzahl von 46 (10cm) auf 56 (20cm) und dann auf 84 (40cm) Fehler.
In Abbildung 4.4 sieht man, dass die Fehleranzahl mit der Triggergröße abgenommen hat.

4.7.3. Umfrage

Der durchschnittliche SUS[1]-Wert beträgt 70%
Frage 1: Für welches Einsatzgebiet könnten Sie sich das System vorstellen?

Zuhause	Bei der Arbeit/Uni	Unterwegs	In der Forschung	In der Lehre
14	5	0	11	9

Tabelle 4.5.: Einsatzgebiete, in denen sich die Probanden das System vorstellen können

Außerdem wurden genannt:

- Im Auto

[1] http://www.measuringusability.com/sus.php

- Bei Präsentationen zum Folien steuern

Frage 2: Wie lange, denken Sie, können Sie sich an die Position der Trigger erinnern?

1 Stunde	1 Tag	1 Woche	1 Monat	Länger
2	4	1	1	7

Tabelle 4.6.: Schätzung der Probanden, wie lange sie sich and die Positionen der Trigger erinnern können.

Frage 3: An wieviele Trigger können sie sich morgen noch erinnern?

1 Trigger	2 Trigger	3 Trigger	4 Trigger	5 Trigger	6 Trigger	7 Trigger
1	0	0	7	1	1	8

Tabelle 4.7.: Schätzung der Probanden, an wieviele Trigger sie sich morgen noch erinnern können

Frage 4: Hat Ihnen eine Funktionalität gefehlt? Wenn ja, welche?
Hier wurden unter anderem eine Demo, ein Helfer, der merkt, wenn man einen Trigger nicht wiederfindet, die Möglichkeit, die Triggergröße nachträglich ändern zu können sowie eine Seitenansicht angegeben. Die detaillierten Antworten finden sich im Anhang A.2

Frage 5: Was hat Ihnen an unserem System nicht gefallen?
Hier wurden auch einige Punkte genannt, unter Anderem die vielen Fehleingaben, dass die Tiefe des Raumes mit einbezogen wird, dass im Wizard keine Körperteile vorgewählt sind und dass die GUI nicht intuitiv genau sei. Die detaillierten Antworten finden sich im Anhang A.2

Frage 6: Was hat Ihnen an unserem System gefallen?
Auch hier gab es einige Punkt, unter Anderem die hardwarelose Bedienung, dass der ganze Raum genutzt werden kann, die intuitive Bedienung und der Spaß bei der Bedienung. Die detaillierten Antworten finden sich im Anhang A.2

Frage 7: Möchten sie uns sonst noch etwas mitteilen?
Hier wurden unter Anderem genannt, dass eine Markierung am Boden hilfreich wäre, 20cm dien beste Größe wäre und das Ganze eine tolle Idee sei. Die detaillierten Antworten finden sich im Anhang A.2

4.7.4. Zusammenfassung

Bei Erstbenutzung spielt der Lerneffekt eine große Rolle. Daher wurde die eine Hälfte der Probanden von Phase 1 zu Phase 3 besser. Wir vermuten, dass bei diesen Probanden die Phase 1 übermäßig schwierig war und damit übermäßig lange dauerte. Daher konnten sie eine Steigerung zur Phase 3 erreichen. Für die andere Hälfte vermuten wir, dass bereits Phase 1 gut lief und ihre Verschlechterung auf die zwischenzeitliche Beschäftigung mit anderen Triggern zurückzuführen ist.
Wie erwartet werden größere Trigger schneller gefunden, allerdings steigt wie erwartet auch die Fehlerzahl erheblich. Bei einer mittleren Größe halten sich beide Eigenschaften die Waage, daher bietet sich eine Größe von ca. 20cm als Idealgröße an.

Der SUS[2] Test zeigt, dass die Usability, mit 70%, im guten Bereich liegt. Das bestätigen auch großteils die Antworten der Probanden. Machen Probanden fanden die Interaktion im dreidimensionalen Raum fragwürdig, allerdings basiert hierauf das Prinzip von ActOnIt. Natürlich kann noch einiges an der Bedienbarkeit von ActOnIt verbessert werden, aber im Großen und Ganzen wurde das System gut angenommen.

[2] http://www.measuringusability.com/sus.php

5. Bewertung und Probleme

In diesem Kapitel werden Probleme der Kinect-Kamera, mögliche Lösungen sowie eine Bewertung der Gesamtsituation vorgestellt.

5.1. Probleme der Kinect-Kamera

Mit einer Kinect-Kamera ist es unmöglich ein vollständiges dreidimensionales Bild aufzunehmen. Sie erkennt zwar Bewegung in allen drei Ebenen, doch kann sie eben nicht hinter ein anderes Objekt sehen. Somit ist es nicht möglich zu erkennen was zum Beispiel hinter einem Tisch passiert. In Abbildung 5.1 zeigt der maximale Aufnahmewinkel den maximal aufnehmbaren Bereich und der mögliche Aufnahmewinkel den Bereich, der wahrgenommen wird.

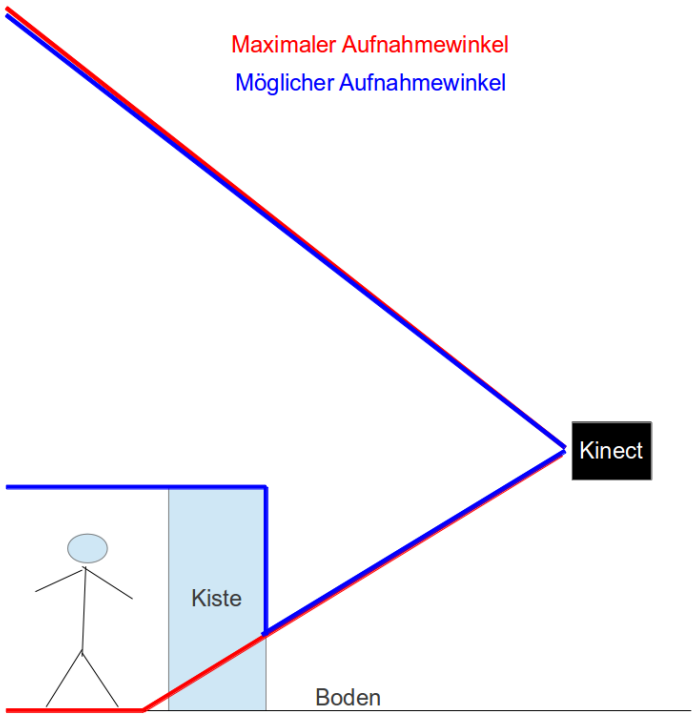

Abbildung 5.1.: Schematische Darstellung, was von der Kinect aufgenommen werden kann und was verdeckt wird.

Dazu kommt, dass die Kinect-Kamera nur einen beschränkten Winkel aufnehmen kann, sie sieht zum Beispiel nicht, was sich hinter ihr befindet. Zum dritten ist es der Kinect-Kamera im Moment nur möglich zwei Personen gleichzeitig zu erkennen

5.2. Mögliche Lösungen

Um ein vollständig dreidimensionales Bild aufnehmen zu können, benötigt man mehr als eine Kinect-Kamera. Befindet sich eine zweite Kamera im 90-Grad-Winkel zur ersten kann sie wahrnehmen, was aus Sicht der ersten Kamera hintereinander geschieht. Durch Kombination aller Daten ließe sich so ein vollständiges dreidimensionales Bild errechnen. Dennoch kann es weiterhin Bereiche geben, die von keiner der beiden Kameras eingesehen werden kann. Um wirklich alle Bereiche eines Zimmers erfassen zu können, bräuchte man also mehrere Kameras. Damit wäre auch das Problem mit dem beschränkten Winkel gelöst. An dem Problem, dass zur Zeit nur zwei Personen gleichzeitig erkannt werden kann, lässt sich momentan nichts ändern, allerdings ist es sehr wahrscheinlich, dass sich die Technik in naher Zukunft dahingehend verbessert.

5.3. Vor- und Nachteile des Systems

5.3.1. Vorteile

- Laufwege werden verkürzt, da man sich Trigger an jede beliebige Stelle definieren kann
- Routineaktionen kann man (wie Shortcuts) per Bewegung starten
- Kräftezehrende Aktionen können über einen Wink ausgeführt werden

5.3.2. Nachteile

- Man muss sich merken, wo die Trigger liegen und mit welchem Körperteil sie aktiviert werden
- Trigger können sich überschneiden und ungewollte Kettenreaktionen auslösen

6. Zusammenfassung und Ausblick

Bei unserer Fachstudie ist ein interessantes und erweiterbares Programm heraus gekommen. Die Studie zeigt, dass das Programm gut benutzbar ist, auch wenn bei der Usabillity noch manches verbessert werden kann. Insgesamt lässt sich sagen, dass die Zukunft den Ubiquitären Systemen gehört und unser Programm ein erster Schritt in diese Richtung ist.

6.1. Ausblick

Da es zur Zeit nur sehr wenige vollständig vernetzte Haushalte gibt, ist es schwer, sich den Nutzen eines solchen Systems vorzustellen. Doch stellen wir uns mal vor, man könnte mit Gesten sämtliche Haushaltsgeräte steuern. Kein lästiges Tasten nach dem Lichtschalter mehr, kein kraftvolles Hochziehen des Rollladens, kein Suchen nach der Fernbedienung. Anstatt im Dunkeln nach dem Lichtschalter zu suchen, legt man einfach seine Hand neben die Tür. Den Rollläden lässt man über einen einfachen Wink hoch und runter fahren. Und der Fernseher erkennt per Fingerzeig, dass er den Ton leiser stellen soll.
Und für all das muss nicht jedes Gerät selbst die Hardware mitbringen, sondern ein System kann alles steuern. Und der Benutzer ist dabei völlig frei von Zwängen, da er selbst bestimmen kann, von wo und wie welche Aktion ausgeführt werden soll.

A. Ein Anhang

A.1. Text für die Studie

Bei unserem Programm werden Actions mit Triggern verbunden. Trigger sind dreidimensionale Würfel, die an einer beliebigen Position vor der Kamera definiert werden. Sie werden durch vorher festgelegte Körperteile ausgelöst. Bewegt man ein entsprechendes Körperteil in den Bereich des Triggers, so wird die dazugehörige Action ausgelöst. Actions sind hier Programm oder Scripte auf dem PC.
Die Studie gliedert sich in drei Phasen.

Phase 1

Bitte erstellen Sie zuerst drei Actions, die die Zahlen eins bis drei ausgeben. , ich werde Ihnen dabei behilflich sein.
Nun erstellen Sie bitte drei Trigger mit der Größe von 10 cm.
Jetzt verbinden Sie bitte jeweils einen Trigger mit einer Action, merken Sie sich dabei, welcher Trigger welche Action auslöst.
Sie sollen nun die drei Actions in aufsteigender Reihenfolge aktivieren. Davon werde ich die Zeit messen.

Phase 2

In dieser Phase gibt es bereits vier Trigger. Sie müssen nur deren Position festlegen. Die Trigger aktivieren in einem Musikprogramm den letzten oder vorherigen Track oder stellen die Lautstärke ein.
Bitte erstellen Sie jetzt die Trigger mit der Größe 10. Ich werde Ihnen nun einige Aufgaben stellen und jeweils die Zeit messen:

- Schalten Sie zwei Lieder nach vorne
- Schalten Sie ein Lied zurück
- Stellen Sie die Lautstärke um eins nach oben
- Stellen Sie die Lautstärke um zwei nach unten
- Schalten Sie zuerst ein Lied nach vorne und machen Sie danach zweimal lauter

- Machen Sie einmal leiser und schalten Sie danach zwei Lieder zurück.

Ändern Sie jetzt die Größe der Trigger auf 20cm.
Die Aufgaben sind die gleichen

Ändern Sie die Triggergröße nun auf 40cm.

Phase 3

Nun stehen wieder die Trigger aus Phase 1 zur verfügung. Bitte aktivieren Sie die Trigger in der richtigen Reihenfolge, ich werde die Zeit stoppen.

A.2. Ergebnisse der Umfrage

- Demo/Test vordefiniert und automatisch
- Helfer, falls das System merkt, dass man einen Punkt nicht wieder findet
- Abstand für Bounding-Box nachträglichen ändern. Körperteile in Wizard vorauswählen
- Ein Signal, beim Ausführen einer Aktion. (Evtl. Ton)
- Kopf als Triggerpunkt/Kopf in der Darstellung fehlt
- Seitenansicht, um die Tiefeninformationen besser zu sehen

Frage 5: Was hat Ihnen an unserem System nicht gefallen?

- Relativ viel Kraftaufwand/Bewegung um eine Aktion auszuführen. Die Trigger sind nicht für alle Personen gleich gut wählbar.
- Praxistauglichkeit
- Viel zu leichte Fehleingaben
- Man könnte etwas mehr beschreibende Texte bei den Steuerelementen hinzufügen.
- Das er die Tiefe mit berechnet
- Das der Abstand zur Kamera für das Auslösen des Triggers passen muss. Cool wäre es, wenn das alleine durch Haltung der Körperteile klappen würde.
- Ermüdet auf Dauer die Muskeln, wenn Triggerpunkte zu hoch (über Schulterhöhe) gesetzt werden. Kamera und damit mögliche Aufzeichnung.
- Kein visuelles Feedback während der Bedienung der Drittsoftware, UI könnte intuitiver sein.

- Stellenweise hätte man die GUI intuitiver machen können - zum Beispiel das ersichtlicher wird, dass der Wizard pro Trigger ist
- Der Wizard startet nicht mit dem eingestellten Körperteil voreingestellt.
- Die Positions-Boxen waren etwas zu klein, es war schwer die exakte Position zu finden.
- Die räumliche dritte Dimension also vorne-hinten
- Nichts (2)

Frage 6: Was hat Ihnen an unserem System gefallen?

- Hardwarelose Bedienung
- Beliebig platzierbar und einstellbarer Radius
- Albernheiten vor der Kinect
- Hat Spaß gemacht
- Das man ganze Programme steuern kann. Vor allem bei Komponierprogrammen öffnet dies ungeahnte Möglichkeiten
- Bedienung ohne spezifische Eingabegeräte (Tastatur/Maus o.ä.)
- Das Konzept allgemein
- Endlich Bewegung beim =(
- Relativ intuitive Bedienung
- Die Musik. ;-) Interessante Interaktionsmöglichkeit
- Audiofeedback bei getriggerten Actions
- Es funktioniert recht gut, solange man die Trigger mit sinnvollen Größen, Positionen und Aktionen anlegt
- -
- Es ist eine gute Idee Systeme durch Körperbewgung zu bedienen

Frage 7: Möchten sie uns sonst noch etwas mitteilen?

- Die Software scheint für die Bedienung im Sitzen evtl. gut zu haben
- Gute Idee
- Markierungen für die Füße am Boden wären hilfreich
- Danke für diese Studie. Ich freue mich auf die Marktreife!
- Größe der Trigger war mit 20 cm eigentlich am besten
- Das Strichmännchen bewegt die Beine komisch, sieht bisschen buggy aus

- Ich könnte mir ein Overlay vorstellen, das den Benutzer und die Triggerpostionen anzeigt. Als optionales Hilfefenster
- -
- Ich sehe den Nachteil, dass man eine bestimmte Postion im Raum haben muss, um die Funktion zu aktivieren

Literaturverzeichnis

[GPC11] L. Gallo, A. Placitelli, M. Ciampi. Controller-free exploration of medical image data: Experiencing the Kinect. In *Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on*, S. 1 –6. 2011. doi:10.1109/CBMS.2011.5999138. (Zitiert auf Seite 10)

[Pan12] G. Panger. Kinect in the kitchen: testing depth camera interactions in practical home environments. In *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts*, CHI EA '12, S. 1985–1990. ACM, New York, NY, USA, 2012. doi:10.1145/2223656.2223740. URL http://doi.acm.org/10.1145/2223656.2223740. (Zitiert auf Seite 10)

[PKLLO05] M. G. Petersen, P. G. Krogh, M. Ludvigsen, A. Lykke-Olesen. Floor interaction HCI reaching new ground. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, S. 1717–1720. ACM, New York, NY, USA, 2005. doi:10.1145/1056808.1057005. URL http://doi.acm.org/10.1145/1056808.1057005. (Zitiert auf Seite 10)

Alle URLs wurden zuletzt am 17. 12. 2012 geprüft.

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.

(Verena Käfer Peter Vollmer Niklas Schnelle)