|  |  |
| --- | --- |
|  | Kinect SA |
|  | Evaluation |

Felder werden durch InStep automatisch ausgefüllt. Bei manuellen Einträgen kann mit Doppelklick der Text selektiert und geändert werden Achtung! Dabei dürfen die Textmarken nicht gelöscht werden, da ansonsten die Verweise in der Fusszeile nicht mehr stimmen.

Änderungsnachweis

| Version | Status | Datum | Beschreibung | Autor |
| --- | --- | --- | --- | --- |
| 1.0 | Erstellung | 28.09.2012 |  | Renato Bosshart |

Inhaltsverzeichnis

[1. Anforderungskriterien 5](#_Toc336610626)

[1.1 Umfeld 5](#_Toc336610627)

[1.1.1 Störfaktoren 5](#_Toc336610628)

[1.2 Anforderungen 5](#_Toc336610629)

[1.2.1 Umgebung 5](#_Toc336610630)

[1.2.2 Workflow 5](#_Toc336610631)

[1.2.3 Auswertung Realisierbarkeit 6](#_Toc336610632)

[1.3 Erweiterungen 6](#_Toc336610633)

[2. Bestehende Bedienkonzepte 6](#_Toc336610634)

[2.1 XBoX-Games 6](#_Toc336610635)

[2.1.1 Microsoft Demovideo 6](#_Toc336610636)

[2.2 The Leap 6](#_Toc336610637)

[2.3 Wii 6](#_Toc336610638)

[2.3.1 Glove Pie 6](#_Toc336610639)

[2.4 PS Move 6](#_Toc336610640)

[2.5 Kinect am PC 6](#_Toc336610641)

[2.5.1 Interactive Wall 6](#_Toc336610642)

[2.5.2 FAAST 6](#_Toc336610643)

[2.6 Sixense 6](#_Toc336610644)

[2.7 Lightgun 7](#_Toc336610645)

[2.8 Touchscreens-Devices 7](#_Toc336610646)

[2.9 2D Gestenerkennung via Webcam oder ähnlichem 7](#_Toc336610647)

[2.9.1 PS EyeToy 7](#_Toc336610648)

[2.9.2 Am PC 7](#_Toc336610649)

[2.10 Minority Report 7](#_Toc336610650)

[2.11 3D-Mouse 7](#_Toc336610651)

[3. Gesten 7](#_Toc336610652)

[3.1 Anmeldung 7](#_Toc336610653)

[3.1.1 Arme nach aussen Halten 7](#_Toc336610654)

[3.1.2 Winken 7](#_Toc336610655)

[3.2 Zoom 8](#_Toc336610656)

[3.2.1 Pinch 8](#_Toc336610657)

[3.2.2 Push/Pull 8](#_Toc336610658)

[3.3 Scrollen = Blättern 8](#_Toc336610659)

[3.3.1 Wischen 8](#_Toc336610660)

[3.3.2 Scrollgeschwindigkeit ist abhängig von der Handposition 8](#_Toc336610661)

[3.3.3 Oberkörper bewegen 8](#_Toc336610662)

[3.4 Auswählen 8](#_Toc336610663)

[3.4.1 Stossen 8](#_Toc336610664)

[3.4.2 Spezielle Aktion mit der anderen Hand 8](#_Toc336610665)

[3.4.3 Thumb up 8](#_Toc336610666)

[3.5 Spezialaktionen (spezielle Aktionen) 8](#_Toc336610667)

[3.5.1 Bestimmter Winkel zwischen Körper und Armen 8](#_Toc336610668)

[4. Abbildungs- und Tabellenverzeichnis 8](#_Toc336610669)

# Anforderungskriterien

## Umfeld

Die Software wird im industriellen Umfeld eingesetzt – konkret in einer grossen Halle. Man muss Bildschirme bedienen können, die bis zu 3m in der Höhe angebracht sind. Die Distanz zum User beträgt etwa 2 - 10 Meter. Die Raumbeleuchtung befindet sich an der Decke – normalerweise Neon oder Halogen. Es könnte aber auch Tageslicht von oben oder der Seite hinzukommen.

In der Halle bewegen sich normalerweise mehrere Personen, teilweise auch kleinere Fahrzeuge. Die Arbeiter haben unter Umständen Schutzkleidung an.

### Störfaktoren

Unter Umständen ist die Kinect kleinen Erschütterungen oder Vibrationen ausgesetzt. Durch Maschinen kann es zu Lärm oder Staubemissionen kommen. Ebenso können Wärmequellen in der Halle entstehen und sich allenfalls bewegen. Bei der Beachtung kann es allenfalls zu direkter Sonneneinstrahlung kommen. LED könnte in Zukunft auch als Leuchtmittel eingesetzt werden. Abgesehen von Personen und Fahrzeugen kann es auch andere Objekte haben, die sich bewegen können (z.B.: ein Kran).

## Anforderungen

### Umgebung

* Ca. 2-10m Distanz zur bedienenden Person
* Personen sollen sich nicht überwacht fühlen (BigBrother-Problem)
* Mehrere Personen können zuschauen, es bedient jedoch immer nur jemand gleichzeitig (offen für mehrere Bedienungen)
* Signalisierung der Einsatzbereitschaft des Systems: Sind Inputs möglich? Hat es zu viel Licht? Person zu weit entfernt, etc.
* Anmeldung der zu bedienenden Person (z.B.: vorbeilaufen soll keine Events triggern)
* Automatische und manuelle Abmeldung (z.B.: was passiert beim Schuhebinden)
* Realisierbarkeit mit einer Kinect
* Aktive und passive Nutzer
* Möglichst unabhängig von oben erwähnten Störfaktoren

### Workflow

* Ausführbare Aktionen/Gesten:
  + Blättern
  + Scrollen
  + Klicken
  + Zoomen
* Einfache Gestik: Personen wollen nicht den „Hampelmann machen“ oder sich exponieren.
  + Eindeutig
  + Einfach lernbar
  + intuitiv
* Gesten sollten nach maximal zwei Versuchen erkannt werden, irrtümliche Inputs sollten nicht vorkommen.
* Einfach kalibrierbar
* Workflow soll intuitiv und nicht zu träge sein, jedoch auch nicht zu empfindlich: guter Tradeoff gesucht

### Auswertung Realisierbarkeit

## Erweiterungen

# Gesten

Begriffe:

Push: Eine Bewegung, bei der die Hand schnell mach vorne bewegt wird.

Pull: Die Hand wird von einer Position vorne schnell zum Körper gezogen.

Pinch-Zoom: Durch die Bewegung von beiden Händen zueinander wird herausgezoomt, wenn sich die Hände auseinanderbewegen wird hinengezoomt.

Push and Pull Zoom: Durch das heranziehen eines Objekt wird es grösser, durch das wegstossen kleiner.

Wischen: eine schnelle Handbewegung nach links oder rechts, die Bewegung kommt aus dem Ellbogen, nicht nur aus dem Handgelenk.

Joystick: Es wird die relative Handposition zu einem definierten Punkt des Körpers verwendet. Darum herum gibt es eine Deadzone, in der nichts passiert. Ausserhalb von dieser wird in diese Richtung gescrollt. Die Geschwindigkeit ist abhängig von der Distanz.

Armwinkel: Der Winkel der zwischen Hand, Schulter und Hüftgelenk aufgespannt wird.

Winken: Hin und her bewegen der Hand, die sich oberhalb des Kopfes befinden muss. Die Bewegung kommt aus dem Ellbogen.

Ausgestreckt: Schulter, Ellbogen und Hand bilden eine Linie. Der Arm zeigt vom Körper weg.

Körper lehnen: Eine Position, wobei der Oberkörper nach vorne, hinten, links oder rechts geneigt ist. Die Körperachse ist dabei gekrümmt in dieser Richtung.

Handgeste: Eine Geste, die nur von der Hand ausgeführt wird. Diese Gesten können wir mit unserer Ausgangslage nicht erfassen. Daher werden wir nicht weiter darauf eingehen.

Laserpointer: Durch Verlängerung einer Controllerachse oder Körperteils erhält man am Durstosspunkt durch die Bildebene einen Punkt wo ein Cursor dargestellt wird.

Absolutes Scrolling: Ein Punkt auf der Bildebene wird fixiert. Das Bild wird genau dem nachfolgenden Bewegungsmuster folgen und der Cursor(falls vorhanden) wird sich noch auf dem gleichen Bildpunkt befinden.

Scrolling mit Momentum: Das Bild wird beim Loslassen mit der gleichen Geschwindigkeit weiterbewegt. Mit einer Dämpfung wird es während einem Zeitinterwall abgebremst.

## Benutzte Gesten anderer Produkte und Projekte

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Device | Eigenheiten und Details | Auswahl | Zoom | Scrolling | Sonstiges |
| Xbox mit Kinect | Individuell für jedes Game, die meisten Games haben Tutorials. Kein allgemeines Konzept | Meist Push, teilweise Handerkennung | 2-Hand Pinch | Wischen, teilweise Joystick | Zusätzliche Aktionen mit z.B. bestimmtem Armwinkel Anmeldung mit Winken |
| Kinect am PC | Noch keine kommerzielle Software, meist spezifische Prototypen oder Emulatoren | individuell |  |  |  |
| Interactive Wall | 3 Beamer, 1 Kinect  Media Center | Push | - | Wischen | Anmeldung mit ausgestreckten Armen, 2 Cursor pro User |
| FAAST | Maus und Tastatur Emulator | Arm strecken | - | Körper lehnen oder Handposition |  |
| The Leap | PC Steuerung, auch mit Emulation, hohe Genauigkeit | Individuelle Handgeste | 2 Finger Zoom oder Push+Pull | Wischen | Noch nicht erhältlich |
| Sixense | Sehr genau, kurze Distanz, zwei Controller, nur Emulation | Button | Button | Button | Laserpointer |
| Glove Pie | Wii Controller am PC | Button | - | - | Nicht zuverlässig |
| Webcam | Objekt oder Handtracking am PC | Handgeste | - | Wischen | Ungenau, störungsempfindlich |
| 3D-Maus | Keine Gesten | Button | Push+Pull | Seitlich bewegen |  |
| Minority-Report | Handgesteuerte Navigation im 3D Raum. Aktionen mit Gesten | Handgeste | 2 Hand Pinch oder Handgeste | Handposition | Kleine Gesten |
| Wii | Ein Controller pro Spieler | Button | - | - | Laserpointer |
| PS Move | Ein Controller pro Spieler | Button | - | - | Genau und schnell |
| Eye Toy | Kollisionsdetektion mit Webcam | - | - | - |  |
| Light-Gun | Point and Click | Button | - | - | Keine Positionsdaten |
| Touchscreen | Weit verbreitet | Klick | 2-Finger Pinch | Absolutes Scrolling | Nur 2D |

## Mögliche Gesten

### Anmeldung

Diese Geste ist sehr wichtig, um dem System mitzuteilen, dass man es jetzt bedienen möchte. Es ist auch schwierig dafür eine gute Geste zu finden, die dem User klar ist. Es ist für einen User ungewohnt, dass er einem System signalisieren muss, dass er interagieren möchte. Alle anderen Inputsysteme reagieren sofort auf einen Input.

### Arme nach aussen Halten

Diese Geste ist ziemlich eindeutig, was Fehlaktivierungen minimieren wird. Der Nachteil ist, dass diese Geste relativ viel Platz braucht, was andere Personen behindern könnte. Zudem braucht es eine Information, da diese Geste unintuitiv ist.

### Slide to Unlock

Diese Geste wird 99% der User bekannt sein, es braucht keine Anleitung dafür. Dafür ist die Erkennung schwieriger, da die Position der Hand nicht klar definiert ist und es wird häufig zu irrtümlichen Aktivierungen kommen. Dafür braucht diese Geste wenig Platz. Apple hat darauf ein Patent, man müsste abklären, wie allgemein das formuliert ist.

### Winken

Diese Geste ist bei den meisten Systemen umgesetzt und ziemlich selbsterklärend. Ein kleiner Text: „Winken zum bedienen“ wird es allen Usern klar machen. Die Erkennung dieser Geste ist etwas einfacher als Slide to Unlock, aber auch hier können selten Fehlaktivierungen vorkommen. In den ersten Wochen des Einsatzes könnte es bei den Mitarbeitern zu Missverständnissen kommen, wenn sie eine Person winken sehen, da dies eine übliche Geste zwischen Personen ist um Aufmerksamkeit zu erlangen. Das exponiert den User und lenkt andere ab in einer grossen Halle. Nach einer gewissen Zeit wird dieses Problem aber verschwinden.

## Zoom

Zoomen ist etwas weit verbreitetes, hier gibt es eigentlich nur zwei intuitive Gesten.

### Pinch

Pinch Zoom ist auf Touchscreens sehr stark verbreitet und dürfte daher jedem User klar sein. Durch den Umstand, dass die Hände in sehr viele Richtungen zueinander und auseinander bewegt werden können, ist die Erkennung hierbei schwieriger und könnte mit anderen Gesten interferieren. Es wird auch zu Fehlaktivierungen kommen, was jedoch nicht kein Problem darstellen wird, da es sich nur um kleine Zoomänderungen handeln wird. Unklar ist ebenfalls, wann die Zoomfunktion aktiviert werden soll und wann der User seine Hände bewegt, damit er nachher eine Zoomfunktion auslösen kann. Zudem werden für diese Geste beide Hände benötigt, in unserem Umfeld wäre eine einhändige Bedienbarkeit sicher ein Vorteil.

### Push/Pull

Je nach Darstellung ist auch diese Geste sehr intuitiv und für jeden User logisch. Diese Geste ist sehr einfach zu erkennen und eindeutig. Dadurch wird es zu wenigen Fehlaktivierungen kommen. Jedoch ist auch hier unklar, wann der User zu einer solchen Bewegung ansetzen will und wann er sie wirklich ausführen will. Diese Geste kann jedoch nur mit einer Hand bedient werden.

Wenn man diese Geste verwendet kann man logischerweise Stossen und ziehen nicht mehr für eine Auswahl verwenden.

## Scrollen = Blättern

### Wischen

Diese Geste wird praktisch bei allen anderen Projekten verwendet. Da man die Geschwindigkeit der Bewegung analysieren kann, werden einerseits Fehlaktivierungen minimiert, andererseits kann man die Distanz und die Geschwindigkeit auf die Aktion übertragen.

### Joystick

Diese Bewegung ist nicht ganz so intuitiv wie andere, dafür hat sie den Vorteil, dass man nicht „nachgreifen“ muss. Zudem ist bei dieser Bewegung der Platzbedarf und der Bewegungsradius grösser. Je nach Gestaltung des GUIs ist auch diese Geste intuitiv und sollte allen Nutzern klar sein.

### Oberkörper bewegen

Diese Geste ist nicht intuitiv und braucht sicher eine Anleitung. Zudem ist sie für unser Umfeld mit spontanen Nutzern nicht wirklich geeignet.

## Cursor bewegen

### Joysick

Hierbei wird die Cursorfunktion und das Scrolling kombiniert. Das ist stabil erkennbar und hat keine Interferenz mit anderen Gesten. Es wird den Usern intuitiv klar sein.

### Zeigen

Dabei wird mit dem Unterarm auf einen Punkt gezeigt. Das ist sicher ungenauer in der Erkennung und benötigt wahrscheinlich eine Kalibrierung für jede neue Session. Zudem wird die Nutzung der Geste „Push“ verunmöglicht. Für den User ist diese Geste sehr intuitiv.

### 2D Mapping der Handposition

Diese Geste ist für User gewöhnungsbedürftig und nicht sehr stabil erkennbar. Dafür interferiert sie nicht mit anderen Gesten und es wäre möglich, pro Hand einen Cursor darzustellen.

## Auswählen

Diese Geste ist sehr wichtig, jedoch nicht ganz einfach umzusetzen, da es bei anderen Technologien dafür immer einen Button oder eine intuitive Lösung gibt. Das ist bei unserer Lösung nicht der Fall. Durch die grosse Distanz wird es zudem unmöglich Handgesten zu erkennen. Deshalb ist es wichtig beim finalen Programm darauf zu achten, dass möglichst wenige Selektionen gemacht werden müssen.

### Stossen

Diese Geste ist bei anderen Projekten am häufigsten umgesetzt und dürfte dem User intuitiv einigermassen klar sein. Dafür ist es bei dieser Bewegung sehr wahrscheinlich, dass währenddessen der Cursor bewegt wird, was zu einer Fehlaktion führen wird.

### Spezielle Gesten mit der anderen Hand

Hier sind verschiedene Gesten denkbar z.B. Stossen, Winken oder ausstrecken. Dadurch werden Fehleingaben minimiert, jedoch ist das einiges weniger intuitiv und benötigt zudem beide Hände.

### Grab

Dabei wird ein Objekt gepackt und zu sich gezogen. Das ist etwas einfacher zu erkennen, Cursorverschiebungen werden weniger häufig vorkommen. Je nach Gestaltung des GUIs ist diese Geste intuitiv klar.

### Nicken

Diese Geste ist intuitiv relativ klar und sollte gut zu erkennen sein. Fehlaktivierungen sind denkbar, wenn sich der User gerade mit jemand anderem unterhält. Das kann jedoch anhand der Blickrichtung korrigiert werden.

### Timer

Wenn der User seinen Cursor nicht bewegt, beginnt ein sichtbarer Timer abzulaufen. Wenn dieser abgelaufen ist, gilt die Selektion, wenn er bewegt wird. Problematisch ist hierbei eine saubere Kalibrierung. Ebenso muss erkannt werden, wann der User etwas zeigen möchte und wann er wirklich etwas selektionieren möchte.

### Thumb up

Eine sehr intuitive Geste, jedoch nicht machbar mit Kinect auf diese Distanz.

## Spezialaktionen (spezielle Aktionen)

Solche sind für unser Projekt nicht nötig, ausser man würde sie für die Anmeldung einsetzten. Dadurch haben wir uns nicht auf solche konzentriert bei der Recherche.

### Bestimmter Winkel zwischen Körper und Armen

Diese Geste ist nicht besonders intuitiv und braucht eine Anleitung. Dafür ist sie gut zu erkennen.

# Abbildungs- und Tabellenverzeichnis