NetEye - iPhone gesteuerte Kamera

Andreas Fischer
Interaktive Medien/FH Augsburg
Fritz-Zubeil-Straße 25
14482 Potsdam
afischer@gmx.com

Katja Loos
Interaktive Medien/FH Augsburg
Friedensstraße 26
07985 Elsterberg
katjaloos@gmx.net

ABSTRACT
Im Projekt wurde eine iPhone Application entwickelt, die eine bewegliche Webcam steuert, den Stream der Kamera sowie deren Position anzeigt. Die Steuerung wird dabei aus Daten des Beschleunigungssensors, sowie des integrierten Kompass erzeugt. Ziel war es eine intuitive Bewegung der Kamera zu ermöglich, die der Bewegung des Nutzers bei Besuch des Standorts der Webcam nachempfunden ist. Das heißt, in der realen Welt dreht sich ein Betrachter im Kreis um seine Umgebung komplett zu erfassen. Mit der Software tut er das selbe mit dem Gerät und sieht ein Videobild, was dem entspricht, was er bei Besuch selbst sehen würde.

Categories and Subject Descriptors
H.5.2 [**User Interfaces**]: Graphical user interfaces (GUI),

H.3.4 [**Systems and Software**]: Performance evaluation (efficiency and effectiveness);

General Terms
Performance, Algorithms, Measurement

Keywords
Interaktion, IP Webcam, iPhone App, IPhone Sensoren

1. MOTIVATION
Es gibt oft wichtige persönliche Momenten wo man nicht da sein kann, weil man gerade auf Arbeit ist oder sogar auf einem Geschäftstermin am anderen Ende der Welt.

Mit NetEye kann man innerhalb weniger Minuten da sein und trotzdem live miterleben wie zum Beispiel das Kind zuhause die ersten Schritte macht. Die Kamera kann sofort mitbewegt werden wenn sich die andere Person aus dem sichtbaren Bereich der Kamera heraus bewegt.

Die Kamera und das iPhone müssen nur mit dem Internet verbunden sein. Aber WLAN gibt es ja sogar in Cafès, an Bahnhöfen und in Gebäuden sowieso.

EyeNet könnte auch bei Meetings über weite Distanzen eingesetzt werden oder wenn man im Urlaub ist und zuhause schauen möchte ob noch alles okay ist.

Außerdem ist die App eine interessante und in vielen anderen Situationen brauchbare Software um zum Beispiel auch einfach nur mit Freunden über das Internet zu reden.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
NetEye, June 27, 2010, Augsburg, Bavaria, Germany.
Copyright 2010 ACM 1-58113-000-0/00/0010…$5.00.

2. Verwandte Arbeiten
Es gibt schon einige Apps für das iPhone die den dort eingebauten Beschleunigungssensor und den Sensor für den Kompass verwenden. Mit dem Beschleunigungssensor kann das iPhone als Wasserwaage benutzt werden. Die App iLevate zum Beispiel kann auch mit Hilfe des Beschleunigungssensor messen wie hoch ein iPhone geworfen wird und die Daten dann speichern und mit einander vergleichen.

Aber dagegen existiert noch keine App die mit den Sensoren eine externe Kamera steuert.

3. ANWENDUNGSSZENARIO & Use Cases

3.1 Besprechung
Änderung des Ausschnitts der Camera, die sich nicht direkt beim Betrachter des Streams befindet, mit Hilfe des iPhones. Um sich im gesamten Raum umsehen zu können, wenn zum Beispiel bei einer Besprechung nicht alle Teilnehmer in den sichtbaren Bereich der Camera passen.

3.2 Definition der Use Cases
USE-CASE 1

Ziel:

Camera nach oben bewegen

Primärer Akteur:

iPhone Benutzer

Sekundärer Akteur:

Kamera

Vorbedingungen:

Kamera und iPhone müssen mit dem Internet verbunden sein

beide müssen eine IP besitzen

Ablauf:

Webcam an Computer anstecken

Computer ins Internet einwählen

iPhone ins WLAN einwählen

App starten

IP der Webcam angeben

'Tilt' Button drücken für die Neigung

iPhone nach vorne neigen

USE-CASE 2

Ziel:

Camera nach unten bewegen

Primärer Akteur:

iPhone Benutzer

Sekundärer Akteur:

Kamera

Vorbedingungen:

Kamera und iPhone müssen mit dem Internet verbunden sein

beide müssen eine IP besitzen

Ablauf:

Webcam an Computer anstecken

Computer ins Internet einwählen

iPhone ins WLAN einwählen

App starten

IP der Webcam angeben

'Tilt' Button drücken für die Neigung

iPhone nach hinten neigen

USE-CASE 3

Ziel:

Camera nach links bewegen

Primärer Akteur:

iPhone Benutzer

Sekundärer Akteur:

Kamera

Vorbedingungen:

Kamera und iPhone müssen mit dem Internet verbunden sein

beide müssen eine IP besitzen

Ablauf:

Webcam an Computer anstecken

Computer ins Internet einwählen

iPhone ins WLAN einwählen

App starten

IP der Webcam angeben

'Pan' Button drücken für den Schwenk

iPhone nach links bewegen

USE-CASE 4

Ziel:

Camera nach rechts bewegen

Primärer Akteur:

iPhone Benutzer

Sekundärer Akteur:

Kamera

Vorbedingungen:

Kamera und iPhone müssen mit dem Internet verbunden sein

beide müssen eine IP besitzen

Ablauf:

Webcam an Computer anstecken

Computer ins Internet einwählen

iPhone ins WLAN einwählen

App starten

IP der Webcam angeben

'Pan' Button drücken für den Schwenk

iPhone nach rechts bewegen

USE-CASE 5

Ziel:

Camera nach links oben bewegen

Primärer Akteur:

iPhone Benutzer

Sekundärer Akteur:

Kamera

Vorbedingungen:

Kamera und iPhone müssen mit dem Internet verbunden sein

beide müssen eine IP besitzen

Ablauf:

Webcam an Computer anstecken

Computer ins Internet einwählen

iPhone ins WLAN einwählen

App starten

IP der Webcam angeben

'P+T' Button drücken für den Schwenk/Neigung

iPhone nach links bewegen, gleichzeitig nach hinten neigen

USE-CASE 6

Ziel:

Camera weder auf Ausgangsposition

Primärer Akteur:

iPhone Benutzer

Sekundärer Akteur:

Kamera

Vorbedingungen:

Kamera und iPhone müssen mit dem Internet verbunden sein

beide müssen eine IP besitzen

Camera musste vorher schon bewegt wurden sein

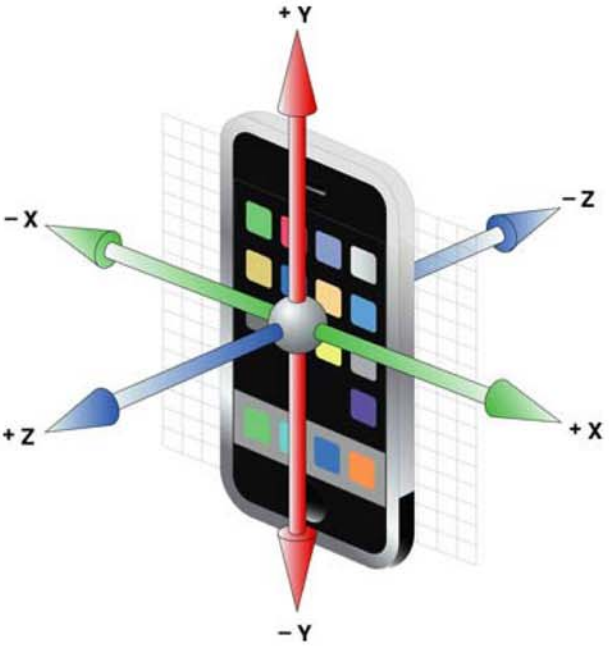
Ablauf:

- Webcam an Computer anstecken
- Computer ins Internet einwählen
- iPhone ins WLAN einwählen
- App starten
- IP der Webcam angeben
- Camera bewegen
- 'Reset' Button drücken

4. ENTWURF EINES PROTOTYPS

4.1 Die Sensoren

Für NetEye haben wir die bereits eingebauten Sensoren des iPhones genutzt.



Der Kompass zeigt den geographischen oder magnetischen Norden bzw. die Abweichung vom Norden. Das iPhone an sich ist die Kompassnadel und die positive y-Achse des Beschleunigungssensors entspricht der Spitze. Kippt man beispielsweise das Gerät entlang der x-Achse und überschreitet den Zenith, verändert sich der Wert um 180°. Der gesamte Wertebereich liegt zwischen 0° und 359°. Die Funktion wird dabei ab einer gewissen Änderung der Gradzahl ausgelöst. Der Schwellenwert liegt bei 7°, da dies der kleinste Wert ist um den die Kamera geschwenkt werden kann. Dreht man das iPhone schneller kommt es zu einer Ungenauigkeit, da bei einer Drehung um 30° nur ein Funktionsaufruf statt findet. Durch Anpassung des zeitlichen Abtastintervals des Sensors kann dies behoben werden. Jedoch muss auf die Kamera Rücksicht genommen werden, da diese nur bedingt schnell Änderungen umsetzt.
Für Bewegungen um die x-Achse kommt der Beschleunigungssensor zum Einsatz. Durch kippen in die eine oder andere Richtung kann eine unterschiedliche Beschleunigung gemessen werden. Bei erreichen eines Schwellenwerts erfolgt dann der Funktionsaufruf. Aprupte Bewegungenkönnen Fehler erzeugen, da eine sehr hohe Beschleunigung auftritt.

4.2 Konzeptueller Aufbau des Prototypen

4.2.1 Server

Die Serverrolle übernimmt in der Konfiguration die Webcam. Sie stellt einen integrierten Webserver zur Verfügung, der Steuerbefehle über das Netzwerk empfängt und intern weiterverarbeitet. Zusätzlich wird die generelle Erreichbarkeit im Netzwerk realisiert.

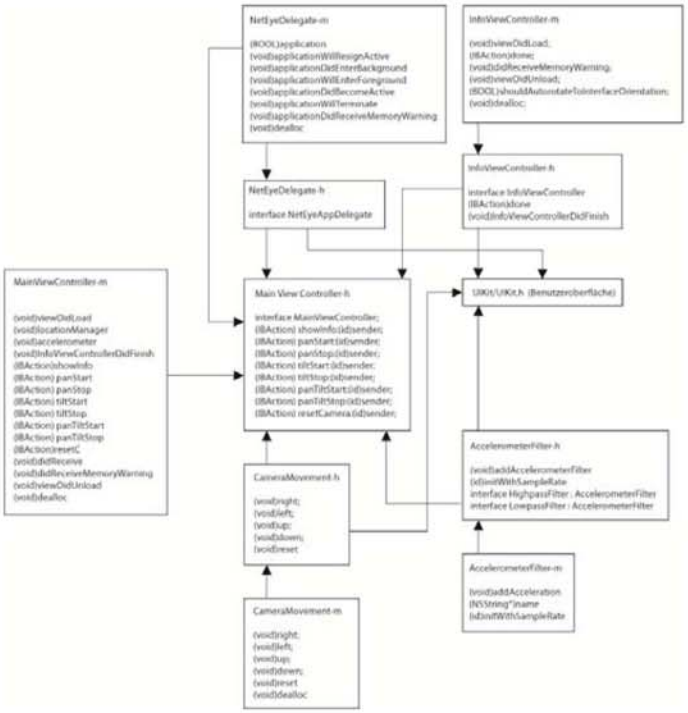


Abb. zeigt Klassendiagramm

4.2.2 Client

Das iPhone agiert als Client. Die durch die Sensorik bestimmten Steuerbefehle werden mittels HTTP Request an den Server gesendet.

5. IMPLEMENTIERUNG

5.1 Hard- und Softwareumgebung

Die Hardware der Entwicklungsumgebung bestand aus einer IP WebCam und einem Computer mit Apple Betriebssystem. Auf dem Computer lief eine iPhone Simulation und die Entwicklungsumgebung Apple Xcode. Für die Benutzeroberfläche haben wir das Apple UIKit verwendet.

Für Programmierung unter Apple wird die Sprache Objective-C und Apple Script benutzt.

5.2 Kern der Implementierung

Der Methode LocationManager ist der wichtigste Teil der Implementierung und wird bei jedem Start ausgeführt. Sie kontrolliert die Bewegungen der Camera.

Die Bewegung der Camera ist standardmäßig nicht fließend, da der eingebaute Motor sich immer nur in 7,73° Schritten weiterbewegt. Um darauf reagieren zu können, gibt es einen Counter der mitzählt wie weit sich das iPhone schon gedreht hat. Sobald der Counter bei 7,73° angelangt ist schickt er ein Event und die Camera bewegt sich.

5.3 Umfang des Prototypen

Im Prototypen haben wir alles geschafft zu realisieren was wir und vorgestellt hatten am Anfang. Es sind sogar noch kleine extra Details für die App hinzu gekommen. Zum Beispiel gibt es einen 'Info' Button und die Bewegung der Camera kann sogar gleichzeitig hochizontal und vertikal erfolgen.

Die Benutzeroberfläche der App besteht aus dem gelieferten Bild der Camera und 4 Button. Einer davon ist der 'Reset' Button um die Camera wieder auf die Ausgangsposition zurück zu bringen. Die andern sind für die Bewegungen. Um die Bewegung durchführen zu können muss der jeweilige Button gedrückt sein. Es gibt einen für 'Pan' (Schwenk), 'Tilt' (Neigung) und einer wenn beides gleichzeitig passieren soll.

6. EVALUATION

Unser Ziel der Evaluation ist die Benutzerfreundlichkeit und Nutzbarkeit der App.

6.1 Ziele der Evaluation und Methodik

Die Entwicklung der App für das iPhone sollte auch zur Folge haben das diese später vielleicht auch an andere iPhone Nutzer verbreitet werden kann. Dazu ist es wichtig auch zu wissen ob mögliche Benutzer damit zurecht kommen wurden oder ob etwas an geändert werden sollte.

Dazu haben wir ein paar Testpersonen die App ausprobieren lassen um zu schauen ob sie damit zurecht kommen.

6.2 Durchführung der Evaluation

Die Testpersonen kamen damit gut zurecht, da die Oberfläche auch sehr einfach und benutzerfreundlich angelegt wurde.

6.3 Interpretation der Evaluationsergebnisse

Da die Oberfläche nur aus wenigen Buttons besteht und es eigentlich gar nicht möglich ist etwas falsch zu machen, ist das Ergebnis das die App benutzerfreundlich ist.

7. ZUSAMMENFASSUNG

Die entwickelte App für das iPhone die mit Hilfe der Sensoren eine externe Camera bewegt war eine sehr interessante Arbei. Mit den Sensoren die bereits standardmäßig im iPhone und jetzt auch im iPad vorhanden sind, kann vieles Realisiert werden was die Interaktion zwischen iPhone und Benutzer nicht nur auf das Drücken von Button beschränkt. Sondern die tatsächliche Bewegung des Telefons kann in die Interaktion mit einfließen und dadurch entstehen viele neue Möglichkeiten für was ein Telefon alles nützlich sein kann.

7.1 Erzielte Ergebnisse

Die vorliegende Studie beschäftigte sich mit der Fragestellung inwiefern die Sensoren des iPhones für Interaktive Anwendung genutzt werden können.

Mit Hilfe von Use-Cases wurde anfangs festgelegt was die Anwendung eigentlich alles können soll.

Der darauf entstandene Prototyp besteht im Wesentlichen aus einer IP Camera und einem iPhone auf dem die Anwendung läuft.

In der anschließenden Evaluation wurde der Prototyp durch Testpersonen hinsichtlich der Benutzerfreundlichkeit untersucht und es wurde festgestellt das die eigentliche Oberfläche der Anwendung benutzerfreundlich ist aber die Eingabe der IP's direkt in der Oberfläche in einer Eingabeaufforderung erfolgen sollte. Ansonsten ist die iPhone App NetEye schon voll funktionsfähig.

7.2 Erweiterungsmöglichkeiten / Ausblick

Um die WebCam mit dem App zu verbinden müssen die jeweiligen IP Adressen eingetragen werden. Das erfolgt im Moment noch direkt über denn Code. Besser wäre es in der Benutzeroberfläche eine Eingabemöglichkeit wäre wo die IP direkt eingeben werden kann um auch Leuten die sich nicht im Code zurechtfinden würden, die Möglichkeit zu geben die Software zu benutzen.

8. DANKSAGUNG

Wir danken Prof. Dr. Thomas Rist von der Fachhochschule Augsburg für die Hilfe.

9. REFERENZEN

[1] Dave Mark, jeff LaMarche 2009
Beginning iPhone 3 Development
Apress, ISBN 978-1-4302-2459-4

[2] Dave Mark, Jeff LaMarche, 2009
More iPhone 3 Development
Apress, ISBN 978-1-4302-2505-8

[3] Mark Dalrymple, Scott Knaster
Learn Objective-C on the Mac
Apress, ISBN 978-1-4302-1815-9

[4] Apple iPhone Developer Center
http://developer.apple.com/iphone/