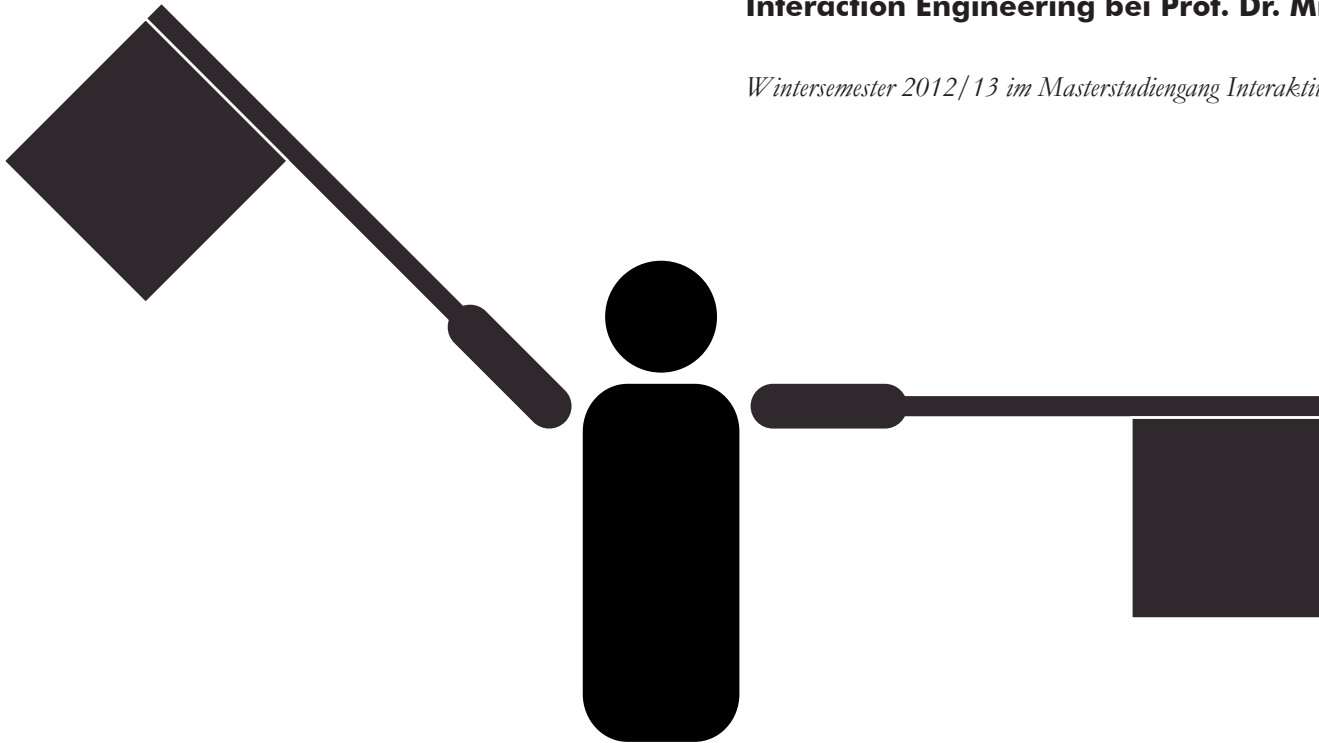


FLAGNECT

Pädagogische Medieninstallation zur spielerischen Erlernung des Winkeralphabets

**Eine Studienarbeit von Moritz Jacobs (MNR 933685) und
Ralph Stachulla (MNR 934342) im Fach
Interaction Engineering bei Prof. Dr. Michael Kipp**

Wintersemester 2012/13 im Masterstudiengang Interaktive Mediensysteme an der Hochschule Augsburg



1. Abstract

FLAGNECT ist eine pädagogische Medieninstallation zum spielerischen Erlernen des sogenannten „Winkeralphabets“ (Bild 1):

„Im internationalen Signalbuch ist festgeschrieben, welche Bedeutung die vom Flaggenwinker gezeigten Stellungen der Winkflaggen (zwei quadratische Flaggen mit rotem Rand und weißem Feld oder diagonal rot/gelb angeordnet am kurzen Stock) haben. Dieses optische Winkflaggenalphabet ermöglicht die (abhörsichere) Verständigung von Schiffsbesatzungen untereinander auf hoher See in Sichtweite. Bevor sich der Funkverkehr und das Radar als Kommunikations- und Navigationsmittel durchgesetzt haben, diente das Winkflaggenalphabet ausschließlich diesen Zwecken.“

— <http://www.code-knacker.de/winkeralphabet.htm>

Der Installationsprototyp besteht aus einem Rechner inkl. Monitor und einer Microsoft Kinect. Je nach Größe der Ausstellung sollte anstatt des Monitors ein Beamer gewählt werden.

Mit Hilfe des räumlichen Trackings der Kinect wird der Benutzer der Installation erkannt und auf eine abstrakte Winkerfigur auf dem Monitor abgebildet. Die abstrakte Figur erfasst lediglich die Bewegung der Arme des Benutzers und gibt je nach dessen Armhaltung den jeweiligen Buchstaben des Winkeralphabets aus und speichert ihn zur Bildung einer Zeichenkette ab.

Als Hilfestellung kann sich der Benutzer vor Spielbeginn eine animierte Figur auf dem Monitor ansehen, die ihm das Winkeralphabet vormacht. Zusätzlich zeigt einem während dem Spielvorgang eine etwas verdunkelte Figur im Hintergrund die exakte Position des jeweiligen Buchstabens an.



Bild 1: Signalgesten beim Winkern auf dem Segelschulschiff Wilhelm Pieck

(Quelle: Wikipedia)

2. Einleitung

FLAGNECT ist als Proof-of-Concept-Idee entstanden, da es uns als idealer Anwendungsfall eines Kinect-Trackings innerhalb einer pädagogischen Installation im Rahmen eines Museums erschien. Solche Installationen begegnen uns in unserem Berufsalltag immer wieder als vom Kunden gewünschte Zusatzleistungen in Ausstellungsgestaltungs-Projekten.

Die Installation soll Besuchern das Winkeralphabet (Bild 2) auf spielerische Art und Weise näher bringen. Es geht nicht darum Nachrichten schnell und einfach zu übertragen, vielmehr soll verdeutlicht werden wie umständlich und unkomfortabel Nachrichten früher vermittelt wurden. Diese Umständlichkeit vermittelt sich dem Besucher durch ungewohnte Armbewegungen und das Suchen der richtigen Geste unter Zeitdruck.

Hier versuchten wir auch Erkenntnisse aus unserem Langvortrag zum Thema „Uncomfortable Interactions“¹ zur Anwendung zu bringen. Die von den Autoren aufgestellten potentiellen Anwendungsgebiete und Möglichkeiten für einen derartig bewusst eingesetzten Diskomfort in Interaktion können wir dadurch noch um einen Punkt erweitern: spielerische „Kniffligkeit“:

Traditionally, HCI has espoused the cause of usability and its sub-goals of time to learn, speed of performance, rate of errors, retention over time, and satisfaction. In exploring the tradeoffs among these, some have argued that deliberately disadvantaging users in some ways may actually bring benefits in others [...]. Inspired by discussions of ambiguity in HCI, a possibility is to emphasise the frustration inherent in unpredictable control and surprising system responses, while the reverse approach of overly precise control may also create discomfort through extreme compliance.

Benford, Greenhalgh, Giannachi, Walker, Marshall, Rodden 2012, S.2; S.6

Es könnte angedacht werden FLAGNECT in einem Schifffahrtsmuseum aufzubauen, da dort die Relevanz zum Ausstellungsinhalt am größten sein wird.

Die Besucher sollen kleine Nachrichten absetzen können, indem sie an einem bestimmten Punkt vor der Installation stehend, die Winkergesten selbst machen. Auf eine Schautafel mit allen Gesten des Winkeralphabets als Anleitung für die Installation soll bewusst verzichtet werden, da das Prinzip des Alphabets spielerisch erlernt werden soll, anstatt durch langes Studieren der Gesten.

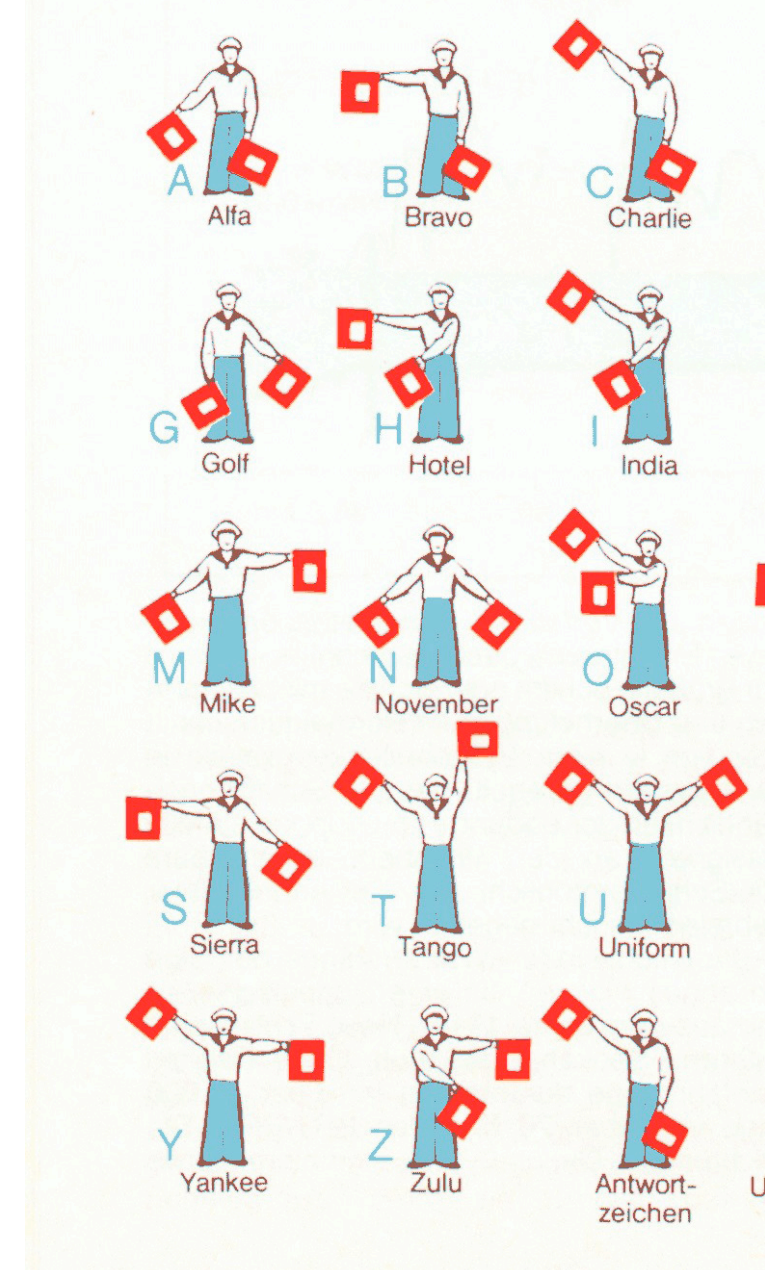


Bild 2: Das Winkeralphabet, Teil einer Schautafel
Quelle: www.wesselhoeft.net

1) Steve Benford, Chris Greenhalgh, Gabriella Giannachi, Brendan Walker, Joe Marshall, and Tom Rodden. 2012. Uncomfortable interactions. In Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems (CHI , 12). ACM, New York, NY, USA, 2005-2014.

3. Systembeschreibung

3.1. Interface

3.1.1. Grundlegendes und Einstieg in die Installation

Gehen wir davon aus, dass der Aufbau der Medieninstallation FLAGNECT in einem Schifffahrtsmuseum steht. Der Monitor oder die projizierte Fläche mit dem User-Interface sollte groß genug sein, damit der Benutzer ohne Schwierigkeiten die Bewegungen der abstrakten Figur, den selbstgeschriebenen Text und die Anweisungen am linken Bildschirmrand erkennen kann.

Da wir davon ausgehen, dass sich mehrere Personen in dem Raum der Installation befinden und das Interface nur für die Benutzung eines Benutzers realisiert ist, sollte ein Bereich abgegrenzt werden in der nur der aktuelle Benutzer stehen kann. So kann gewährleistet werden, dass das Tracking der Kinect optimal funktioniert und es nicht durch andere Personen gestört wird.

Zunächst sieht der Benutzer einen „Demoscreen“ (Bild 3) in dem eine abstrakte Figur das Winkeralphabet in Endlosschleife vorführt. Durch die Textanzeige am unteren Bildschirmrand wird gleichzeitig das Spielprinzip deutlich: Ein Buchstabe wird an die Zeichenkette angefügt, sobald sich das Kreisdiagramm (im Folgenden „Timer“) vollständig gefüllt hat. Solange soll der Benutzer in der jeweiligen Buchstabenposition verharren.

Der *call to action* auf dem Screen im Leerlauf-Modus (kein registrierter Spieler) bittet den User in unserem Prototypen eine beliebige Taste auf der Tastatur zu drücken. Im Falle eines echten Einsatzes in einer Ausstellung, müsste sich der Besucher stattdessen auf eine bestimmte Markierung auf dem Boden stellen, die mit Hilfe eines Drucksensors (Piezo unter einem Teppich z.B.) dem System das Startsignal gibt. Dann springt das Programm aus dem Demomodus und startet das Tracking.



Bild 3: Demoscreen der Anwendung im Leerlauf

3.1.2. „Spielmodus“

Sobald das Tracking erfolgreich gestartet wurde, erscheint eine hellere abstrakte Figur im Vordergrund. Diese Figur bildet nun die Bewegungen des Benutzers auf dem Monitor ab, die dunklere Figur dahinter dient nur noch als Hilfestellung um die optimale Armhaltung des gewünschten Buchstabens anzuzeigen (Bild 4).

Der Benutzer kann jetzt seine Nachricht „schreiben“ und hat im Rahmen des eingebauten Timers die Möglichkeit durch die Buchstaben des Winkeralphabets zu „scrollen“. Erst wenn er in einer bestimmten Position verharrt wird der Buchstabe an die Zeichenkette angehängt (Bild 5).

Auf Leerzeichen, Korrekturmöglichkeit und Zahlen wurde bewusst verzichtet um das Interface nicht unnötig zu verkomplizieren. Es geht schließlich nicht darum eine fehlerfreie Nachricht an die Öffentlichkeit zu schicken, sondern zu erfahren auf welcher unkomfortable und aufwendige Art und Weise die Matrosen früher Nachrichten übermittelten.

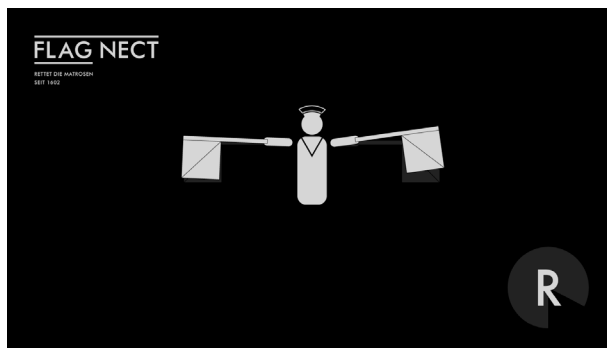


Bild 4: Das Tracking wurde erfolgreich gestartet ...

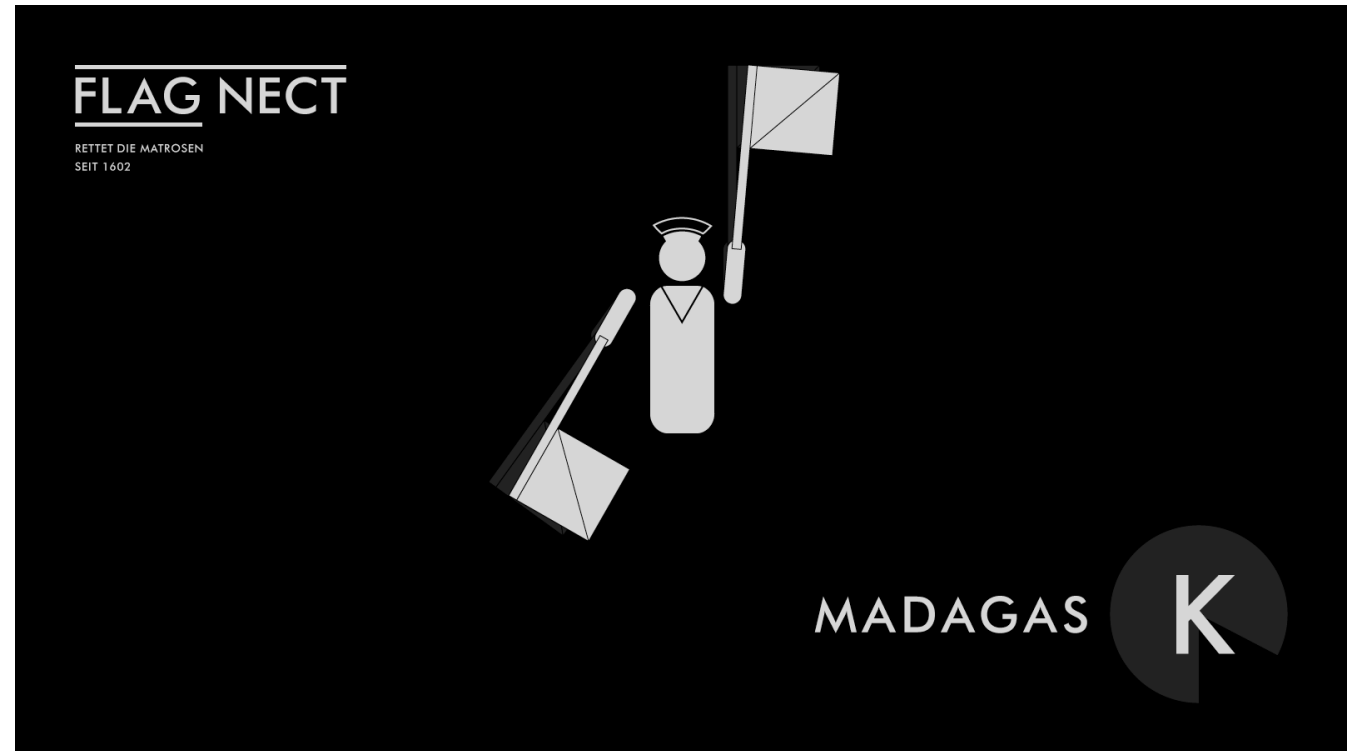


Bild 5: ... und der Benutzer schreibt seine Botschaft

3.1.3. Benutzer unsicher oder in Ruheposition

Hält der Benutzer eine Geste, die nicht im Winkeralphabet vorkommt oder befindet er sich in Ruheposition (beide Arme nach unten gehalten), so wird ein Fragezeichen als Feedback angezeigt. Solange wird das Schreiben der Zeichenkette unterbrochen.

(Bild 6)



Bild 6: Der Benutzer lässt die Arme hängen: Pause

3.1.4. Nachricht beendet oder Tracking verloren

Sobald der Benutzer den trackbaren Bereich der Kinect verlässt oder durch einen Fremdkörper verdeckt wird verschwindet seine abstrakte Abbildung aus der Anwendung und eine Textanzeige bedankt sich für die Nachricht und gibt die Möglichkeit innerhalb von 10 Sekunden doch noch die Nachricht fortzusetzen (Bild 7). Erfolgt dies nicht, wird die Nachricht gelöscht und das Programm springt zurück in den Demomodus.



Bild 7: Der Benutzer hat die Installation verlassen

3.2. Implementierung

FLAGNECT wurde mit Hilfe von SimpleOpenNI¹, Processing/Java (via Eclipse) und OSCeaton² implementiert. OSCeaton ist ein Proxyserver, der die von OpenNI ausgewerteten Skelettdaten der Kinect als OSC-Nachrichten³ auf localhost broadcastet.

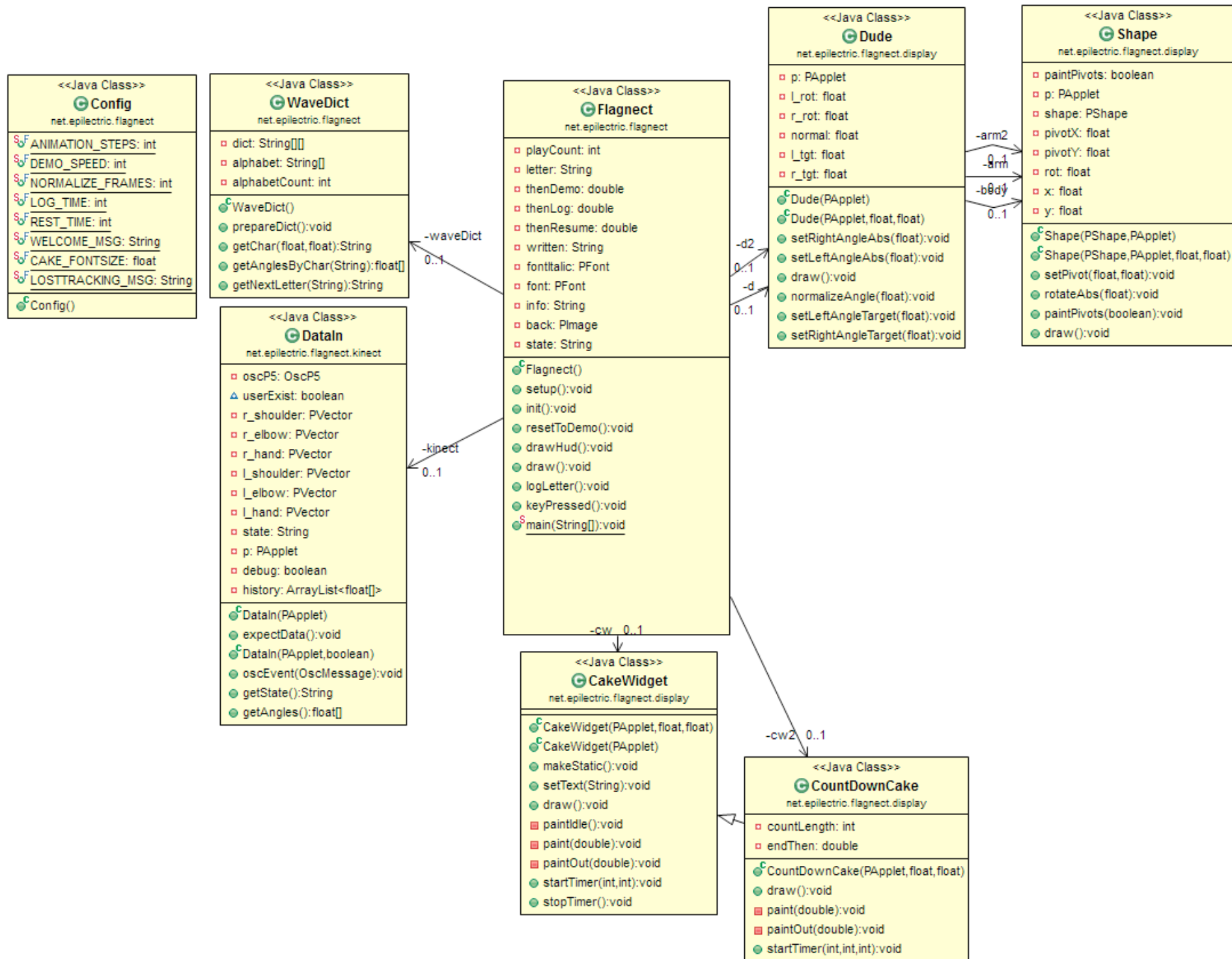
Aus Skelettdaten werden lediglich die Positionsvektoren der beiden Schultern und Hände für eine Winkelbestimmung (in Processing mit `PVector.angleBetween(v1, v2)`) verwendet. Einige Tests haben gezeigt, dass eine Glättung der Messdaten nicht nötig ist, da das direkte Feedback für den User hilfreicher ist, als ein latenter, bereinigter Mittelwert. Daher müssen wir ein optimales Tracking durch äußere Umstände (Abstand zur Kinect, abgesteckter Bereich ohne Störungen durch andere Menschen etc.) sicherstellen.

Die gemessenen Winkel der beiden Arme (zur Senkrechten, von 0° bis 180°) werden dann mit einem Datensatz verglichen, der das Winkeralphabet als zweidimensionales Dictionary enthält (`dict[alpha][beta]`). Die notwendigen Daten (sowie übrigens die Vorlage für die Grafik) haben wir der deutschen Wikipedia entnommen⁴.

Die grafische Darstellung des Interfaces erfolgt mit den gewohnten Möglichkeiten von Processing (PNG und SVG).

Auf der nächsten Seite finden Sie ein ergänzendes Klassendiagramm.

1) <http://code.google.com/p/simple-openni/>
2) <https://github.com/Sensebloom/OSCeaton>
3) <http://opensoundcontrol.org/>
4) <http://de.wikipedia.org/wiki/Winkeralphabet>



4. Pro und Contra

Bei der prototypischen Umsetzung von FLAGNECT haben wir versucht die von uns gewünschte Interaktionsführung möglichst so umzusetzen, wie wir sie von einer echten Museumsinstallation erwarten würden. Leider sind die Funktionsweisen des Kinect-Trackings noch nicht so zu 100% ausgereift, dass man als „Anfänger“ sofort eine saubere, marktreife Implementierung einer Idee erreichen kann, schon gar nicht wenn man so prototypisch arbeitet wie wir hier.

Dennoch funktioniert das Erkennen der Winkergesten erstaunlich gut. Das Suchen von Buchstaben im Alphabet ist gerade so intuitiv, dass man nicht komplett frustriert aufgibt, aber es erfüllt dennoch den Zweck zu zeigen, dass dies eine sehr umständliche und langsame Form der „Telekommunikation“ war.

Nach anfänglichen Schwierigkeiten und „Winkfehlern“ stellte sich bei den meisten unserer Testpersonen nach nur sehr kurzer Zeit ein Gewöhnungseffekt ein, so dass sie schnell in der Lage waren, ganze Wörter zu schreiben.

5. Ausblick

Falls sich ein Kunde für dieses System finden sollte, wären mehrere Punkte noch zu überarbeiten: wie in 3.1.1 bereits erwähnt, sollte ein Drucksensorsystem im Boden entwickelt werden um einen Besucher an der Installation anzumelden. Weiterhin könnte man noch Zeit und Energie in die Optimierung des Trackings investieren, da sich manchmal Jitter in das Tracking einschleichen (Menschen die durchs Bild laufen, unsaubere Armhaltung etc.), die man mit den richtigen Mitteln noch ausbügeln könnte.

Wenn weiterhin noch Zeit und Budget für ausgiebigere Benutzertests vorhanden wäre, könnte man hier noch Methoden aus der Usability-Forschung hinzuziehen. Kleine Benutzertests haben wir mit unseren Freunden und Kollegen während der Entwicklung des Prototyps bereits durchgeführt.

6. Literatur

Steve Benford, Chris Greenhalgh, Gabriella Giannachi, Brendan Walker, Joe Marshall, and Tom Rodden. 2012.

Uncomfortable interactions.

In Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems (CHI ,12). ACM, New York, NY, USA, 2005-2014.

<http://www.mrl.nott.ac.uk/~sdb/research/downloadable%20papers/Uncomfortable%20interactions.pdf>

Winkeralphabet/Semaophore auf Code-Knacker.de

<http://www.code-knacker.de/winkeralphabet.htm>

Winkeralphabet in der deutschen Wikipedia

<http://de.wikipedia.org/wiki/Winkeralphabet>

Winkeralphabet auf Wesselhoeft.de

<http://www.wesselhoeft.net/Winker/Winker.htm>

Verwendete Software und Bibliotheken

<http://code.google.com/p/simple-openni/>

<https://github.com/Sensebloom/OSCeleton>

<http://opensoundcontrol.org/>

<http://www.eclipse.org>

<http://www.processing.org>