# Entwurf

## Änderungsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Änderung | Autor |
| 8.3.2011 | 1.0 | Erste Version des Dokuments | LE |

## Design Entscheide

### Auswahl Kinect Framework

Die Auswahl des Kinect Frameworks ist in einem Projekt, bei dem mit Kinect gearbeitet wird, sehr wichtig, da schliesslich das Projekt mit einem Framework betrieben werden muss. Zu den einzelnen Frameworks gehören auch spezifische Gerätetreiber, welche es verunmöglichen, parallel zwei verschiedene Frameworks einzusetzen. Zur Entscheidung wurde schliesslich eine Nutzwertanalyse durchgeführt.

### Frameworks

#### Framework 1: Kinect for Windows SDK[[1]](#footnote-1)

Das offizielle Kinect Framework von Microsoft wurde kurz vor Beginn dieser Arbeit, im Februar 2012, in der Version 1.0 herausgegeben. Dies ist relativ spät, beachtet man, dass andere Frameworks (OpenNI) schon vor einem Jahr (Ende 2010) veröffentlicht wurden. Entsprechend sind für dieses Framework viel weniger Beispiele und Bibliotheken im Internet zu finden, wobei dafür die Qualität der gefundenen Beispiele und Bibliotheken hoch ist.

Dieses Framework geht durch die Nutzwertanalyse (siehe I.2.3 Nutzwertanalyse) klar als Sieger hervor.

#### Framework 2: OpenNI[[2]](#footnote-2)

Dieses Framework wurde in der Version 1.0.0.23 im Dezember 2010 erstmals freigegeben und konzentriert sich, im Gegensatz zur Microsoft Kinect SDK, nicht nur auf Kinect als Eingabemöglichkeit, sondern allgemein auf Natural User Interfaces (NUI).

Um weitere Geräte anzusprechen und gerätespezifische Funktionen zu implementieren, lässt sich in das Framework zusätzliche Middleware einsetzen. So wird mit NITE[[3]](#footnote-3) von PrimeSense[[4]](#footnote-4) entwickelt, um das Skeleton Tracking durchzuführen.

#### Framework 3: OpenKinect[[5]](#footnote-5) / libfreenect[[6]](#footnote-6)

OpenKinect ist eine Community, die den libreenect Treiber entwickelt. Leider gibt es dafür aber keine erweiterten Funktionen wie Gestenerkennung oder Skeleton Tracking.

### Nutzwertanalyse

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nutzwertanalyse: Auswahl Kinect Framework | | | | | | | |
|  |  | **Framework 1** | | **Framework 2** | | **Framework 3** | |
|  |  | **Kinect for Windows SDK** | | **OpenNI** | | **OpenKinect / libfreenect** | |
| Kriterium | **Gewicht** | **Bewertung** | **Total** | **Bewertung** | **Total** | **Bewertung** | **Total** |
|  | **1 bis 5** | **1 bis 5** |  | **1 bis 5** |  | **1 bis 5** |  |
| Cooperate Support, Weiterentwicklung, Community | 5 | 5 | 25 | 3 | 15 | 3 | 15 |
| Windows Integration und Installation | 3 | 5 | 15 | 3 | 9 | 3 | 9 |
| Linux / Mac Kompatibilität | 1 | 1 | 1 | 5 | 5 | 5 | 5 |
| C# / .NET Framework / Visual Studio Integration | 5 | 5 | 25 | 3 | 15 | 1 | 5 |
| Skeleton Tracking Qualität | 5 | 5 | 25 | 3 | 15 | 1 | 5 |
| Libraries für Gestenerkennung | 3 | 3 | 9 | 5 | 15 | 1 | 3 |
| Record / Replay Funktionalität | 5 | 3 | 15 | 5 | 25 | 1 | 5 |
| Dokumentation | 5 | 5 | 25 | 3 | 15 | 1 | 5 |
| Mit Framework realisierte Beispiele und Libraries (Quantität) | 3 | 1 | 3 | 5 | 15 | 5 | 15 |
| Mit Framework realisierte Beispiele und Libraries (Qualität) | 3 | 5 | 15 | 1 | 3 | 1 | 3 |
| Total Punkte |  |  | **158** |  | **132** |  | **70** |
| Rang |  |  | **1** |  | **2** |  | **3** |
|  |  |  |  |  |  |  |  |
| Bemerkung: Höhere Gewichtungen / Bewertungen sind besser. | | | | | | | |

Tabelle 1 - Nutzwertanalyse

In der Nutzwertanalyse (siehe Tabelle 1 - Nutzwertanalyse) geht das Framework 1: Microsoft Kinect SDK als Sieger vor dem Framework 2: OpenNI hervor.

### Sensitivitätsanalyse

In der Sensitivitätsanalyse wird untersucht, wie stark sich eine Änderung auf das Gesamtergebnis auswirken würde.

Das Framework 3 wird auch bei Änderungen der Bewertung nicht als Sieger hervorgehen.

Zwischen dem Framework 1 und 2 ist der Abstand einiges kleiner. Da das Framework 1 von Microsoft über ein ausgeklügeltes vorhersehendes Skeleton Tracking System[[7]](#footnote-7), eine bessere Dokumentation besitzt, perfekte Windows, Visual Studio, C# und .NET Integration bietet, würde sich dieses Framework trotz Anpassungen an einzelnen Gewichtungen oder Bewertungen gegenüber dem Framework 2 durchsetzen. Demensprechend ist diese Nutzwertanalyse nicht sensitiv gegenüber Änderungen.

### Weiteres

Bei der Nutzwertanalyse wurden zwar möglichst viele nummerisch bewertbare Kriterien untersucht, es fehlt aber noch unser persönlicher Eindruck. Wir würden uns gefühlsmässig klar für das Microsoft Framework entscheiden, und zwar deshalb, weil wir auch sonst überall Microsoft Technologien einsetzen und wir damit rechnen würden, dass andere Frameworks nicht ohne Probleme mit den restlichen Microsoft Technologien zusammenarbeiten könnte.

Zusätzlich wurde in der Bachelorarbeit „Kinect Bodyscanner“[[8]](#footnote-8) von Felix Egli und Michael Schnyder im Abschnitt 3.3.1 Kinect Framework auf Seite 30 beschrieben, dass OpenNI für ihre Arbeit nur eine temporäre Lösung sei. Sie würden sofort zum Kinect Framework von Microsoft wechseln, sobald dieses das Framework fertig entwickelt sei. Auch diese Aussage spricht sich klar für das „Kinect for Windows SDK“ aus.

1. <http://www.microsoft.com/en-us/kinectforwindows/> [↑](#footnote-ref-1)
2. <http://openni.org/> [↑](#footnote-ref-2)
3. <http://www.primesense.com/Nite/> [↑](#footnote-ref-3)
4. <http://www.primesense.com/> [↑](#footnote-ref-4)
5. <http://openkinect.org/wiki/Main_Page> [↑](#footnote-ref-5)
6. <https://github.com/OpenKinect/libfreenect> [↑](#footnote-ref-6)
7. <http://www.cs.dartmouth.edu/~cs104/BodyPartRecognition.pdf> [↑](#footnote-ref-7)
8. <http://eprints3.hsr.ch/180/1/Hauptdokument.pdf> [↑](#footnote-ref-8)