
Inhaltsverzeichnis

1	Einleitung	2
1.1	Anwendungskontext	2
1.2	Projektziel	2
2	Grundlagen	2
3	Materialien	2
3.1	Hardware	2
3.1.1	Rechner 1	2
3.1.2	Rechner 2 ??	2
3.1.3	HTC Vive	2
3.1.4	Ovrvision Pro	2
3.1.5	Ueye 164xLE	3
3.1.6	Leap Motion ??	3
3.2	Marker	3
3.2.1	ArUco Marker	3
3.2.2	Trackingmarker	3
3.2.3	Spielfeldkalibrierung	4
3.2.4	Marker uEye Kalibrierung	4
3.3	Software	4
3.3.1	Unity	4
3.3.2	Visual Studio	4
3.3.3	OpenCV	4
3.3.4	Ovrvision Pro sdk	4
3.3.5	Leap Motion sdk	4
3.3.6	Ueye sdk	4
3.3.7	steam vr	4
4	Grundlagen	5
5	Auswertung	5
5.1	Testing durch unerfahrene Benutzer	5
5.2	Testing durch Architekten	5
6	Zusammenfassung	5

1 Einleitung

1.1 Anwendungskontext

1.2 Projektziel

2 Grundlagen

- Wissenschaftlicher Stand

3 Materialien

3.1 Hardware

3.1.1 Rechner 1

Noch nicht klar.

3.1.2 Rechner 2 ??

Noch nicht klar.

3.1.3 HTC Vive

Bei der HTC Vive handelt es sich um ein Head-Mounted Display, welches von HTC in Kooperation mit Valve [4] produziert wird. Vorgestellt wurde diese am 1. März 2015 im Vorfeld der Mobile World Congress [2].

Die Auflösung des Displays beträgt insgesamt 2160x1200 Pixel, was 1080x1200 pro Auge entspricht. Die Brille bietet ein Sichtfeld von bis 110° bei einer Bildwiederholrate von 90 Hz [1]. Zur Positionsbestimmung im Raum wird die Lighthouse-Technologie von Valve genutzt. Zusätzlich sind neben einem Gyrosensor auch ein Beschleunigungsmesser und ein Laser-Positionsmesser verbaut. Mittels speziellen Game-Controllern wird eine Interaktion mit virtuellen Objekten ermöglicht. Die eingebaute Frontkamera wird für dieses Projekt nicht verwendet. Stattdessen wird auf die Ovrvision Pro zurückgegriffen, die im Folgenden in 3.1.4 beschrieben wird.

3.1.4 Ovrvision Pro

Bei der Ovrvision Pro handelt es sich um eine open-source Stereokamera, welche über USB 3.0 mit dem Rechner verbunden werden kann [3]. Sie ist kompatibel

mit Programmen wie Unity, welches für das Projekt benutzt wurde und in 3.3.1 beschrieben wird.

3.1.5 Ueye 164xLE

3.1.6 Leap Motion ??

[?]

3.2 Marker

3.2.1 ArUco Marker

Da sowohl für die Marker, die für den eigentlichen Trackingalgorithmus verwendet werden, als auch für sämtliche Marker, die zur Kalibrierung des Systems zum Einsatz kommen ArUco Marker verwendet werden, werden diese im Folgenden kurz erläutert.

ArUco Marker bestehen ähnlich wie QR-Codes aus einer zweidimensionalen Matrix, aus schwarzen und weißen Quadraten, die die kodierten Daten binär darstellen. Die ArUco Bibliothek kann für Augmented Reality Anwendungen genutzt werden und basiert ausschließlich auf der OpenCV Bibliothek.

3.2.2 Trackingmarker

Das System umfasst 12 Marker über die das Tracking realisiert wird. Alle Marker stimmen in Form und Farbe, sowie Material und Oberflächenbeschaffenheit überein. Sie sind würfelförmig und haben eine Kantenlänge von 46mm. Die Kanten sind in einem Winkel von 45° angefast. Die Marker bestehen aus Aluminium, welches glasperlgestrahlt ist um eine matte Oberfläche zu erzeugen. Auf die Oberseite des Markers ist mittig ein grünes Quadrat mit einer Kantenlänge von 40mm aufgebracht. Auf diesem ist, ebenfalls mittig, ein 35 mm großer ARUCO-Marker, welcher aus dem *DICT_4X4_50* generiert wurde und einen Rand von einem bit hat. Jeder Marker hat einen einzigartigen ARUCO-Marker, der einer Id von 1-12 entspricht.

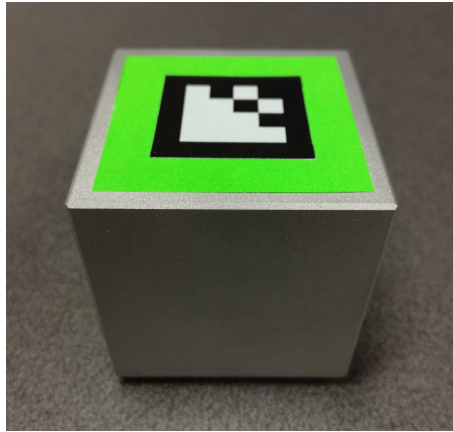


Abbildung 1: Trackingmarker mit der ID ??.

3.2.3 Spielfeldkalibrierung

Marker auf schwarzen Holzplatten von Luke

3.2.4 Marker uEye Kalibrierung

Poster von Vera

3.3 Software

3.3.1 Unity

3.3.2 Visual Studio

3.3.3 OpenCV

3.3.4 Ovrvision Pro sdk

3.3.5 Leap Motion sdk

3.3.6 Ueye sdk

3.3.7 steam vr

4 Grundlagen

5 Auswertung

5.1 Testing durch unerfahrene Benutzer

5.2 Testing durch Architekten

6 Zusammenfassung

- Wissenschaftlicher Stand

Literatur

- [1] HTC Vive. <https://www.vive.com/>. Accessed: 30. November 2016.
 - [2] Mobile World Congress. <https://www.mobileworldcongress.com/>. Accessed: 30. November 2016.
 - [3] Ovrvision Pro. <http://ovrvision.com/>. Accessed: 30. November 2016.
 - [4] Valve Software. <http://www.valvesoftware.com/>. Accessed: 30. November 2016.
-