



---

# Unity 3D Server for CAVE Rendering

Pflichtenheft

Julien Villiger, Daniel Inversini  
V2.0, 18.10.2015

# Inhaltsverzeichnis

|       |   |    |
|-------|---|----|
| 1     | Allgemeines                             | 3  |
| 1.1   | Zweck dieses Dokumentes                 | 3  |
| 1.2   | Lesekreis                               | 3  |
| 1.3   | Ausgangslage                            | 3  |
| 1.4   | Unity                                   | 4  |
| 1.5   | Nvidia Mosaic                           | 4  |
| 1.6   | Infrastruktur                           | 4  |
| 1.6.1 | Inputs und Tracking                     | 5  |
| 1.6.2 | Rendering                               | 6  |
| 1.6.3 | Darstellung im CAVE                     | 7  |
| 2     | Umfang und Ziele der Bachelor Thesis    | 8  |
| 2.1   | Abgrenzungen                            | 8  |
| 2.1.1 | Technische Abgrenzungen                 | 8  |
| 2.1.2 | Weitere Abgrenzungen                    | 8  |
| 2.2   | Voraussetzungen und Ressourcen          | 8  |
| 3     | Funktionale Anforderungen               | 9  |
| 3.1   | Adaption Unity Anwendung für den CAVE   | 9  |
| 3.2   | VRPN Unterstützung                      | 9  |
| 3.3   | Kompatibilität                          | 9  |
| 3.4   | WorldViz Tracking                       | 10 |
| 3.4.1 | Head Tracking                           | 10 |
| 3.4.2 | Wand Tracking                           | 11 |
| 3.5   | Einstellungsmöglichkeiten für 6DoF      | 11 |
| 3.6   | Setup                                   | 11 |
| 3.7   | Demoapplikation                         | 12 |
| 3.8   | Features                                | 13 |
| 3.9   | Zusätzliche Anforderungen               | 13 |
| 3.9.1 | Unity-Output Warping                    | 13 |
| 4     | Nicht funktionale Anforderungen         | 14 |
| 4.1   | Wiederverwendbarkeit                    | 14 |
| 4.2   | Ergonomie                               | 14 |
| 4.3   | Skalierbarkeit sekundäre Kamera         | 14 |
| 5     | Testing                                 | 15 |
| 5.1   | System Tests                            | 15 |
| 5.2   | Usability Tests                         | 15 |
| 5.3   | Szenarien                               | 15 |
| 6     | Administratives                         | 15 |
| 6.1   | Projektorganisation                     | 15 |
| 6.1.1 | Projektteam                             | 15 |
| 6.1.2 | Betreuer                                | 15 |
| 6.1.3 | Experte                                 | 15 |
| 6.2   | Projektplan                             | 16 |
| 6.3   | Projektsitzungen                        | 16 |
| 6.4   | Meilensteine                            | 16 |
| 6.4.1 | Prototyp Unity Plugin                   | 17 |
| 6.4.2 | Tracking                                | 17 |
| 6.4.3 | Unity Plugin / Handbuch / Dokumentation | 17 |
| 6.4.4 | Präsentation                            | 17 |
| 7     | Versionskontrolle                       | 18 |
| 8     | Abbildungs- und Tabellenverzeichnis     | 19 |
| 9     | Glossar                                 | 19 |

# 1 Allgemeines

## 1.1 Zweck dieses Dokumentes

Mit diesem Pflichtenheft wird der Rahmen, die Vorgehensweise und die Ziele der Bachelorthesis dokumentiert.

## 1.2 Lesekreis

Der Inhalt dieses Dokumentes richtet sich in erster Linie an die Betreuer und Experten dieser Arbeit, Prof. Urs Künzler und Herrn Harald Studer, die BFH-TI Abteilung CPVR<sup>1</sup> und an die Studenten, welche diese Projektarbeit durchführen.

## 1.3 Ausgangslage

Das cpvrLab besitzt eine CAVE<sup>2</sup> (Cave Automatic Virtual Environment) Installation mit dem virtuelle 3D-Welten in Echtgrösse über drei Projektionswände und eine Bodenprojektion erzeugt werden können. Alle Projektionsflächen werden dabei mit zwei Projektoren stereoskopisch projiziert, sodass eine nahezu perfekte Raumwahrnehmung entsteht. Die Entwicklung von virtuellen 3D-Welten mit Basis-APIs wie OpenGL<sup>3</sup> oder OpenSceneGraph<sup>4</sup> ist nach wie vor eine zeitraubende und aufwendige Arbeit und jedes Mal eine Einzelentwicklung. Es liegt deshalb nahe, eine High-Level Game Engine einzusetzen, mit der die Entwicklungszyklen vereinfacht und verkürzt werden können. Unity hat sich in den letzten Jahren in diesem Bereich durchgesetzt und ermöglicht es, gratis Spiele zu entwickeln.

Im Rahmen des vorgängigen Projekts, die Projekt 2 Arbeit, wurden verschiedene Methoden geprüft wie eine Integration von Unity in den CAVE erfolgen kann.

In einem ersten Schritt erfolgte eine Prüfung der bereits verwendeten Frameworks (Equalizer<sup>5</sup>, Chromium<sup>6</sup>) und der Software MiddleVR, die den Einsatz von Unity in einem CAVE deutlich vereinfachen sollte. Die dadurch entstehende Abhängigkeit der Software, die Anforderungen an die Unity<sup>7</sup>-Applikationen (etliche Adaptionen am Code waren jeweils notwendig) und die nicht gebrauchte Fülle an Features waren der Grund, wieso eine eigene Umsetzung eines Unity-Plugins erfolgen sollte.

Bezüglich der Infrastruktur wurden folgende Methoden analysiert und bewertet:

1. Mehrere Hosts / Mehrere GPUs / Mehrere Unity Instanzen
2. Ein Host / Mehrere GPUs / Mehrere Unity Instanzen
3. Ein Host / Mehrere GPUs / Eine Unity Instanz
4. Ein Host / Mehrere GPUs / Eine Unity Instanz mit Mosaic<sup>8</sup> Treiber

Basierend auf Prototypen der verschiedenen Systeme und theoretischer Abklärungen konnte sich Variante 4 (Ein Host / Mehrere GPUs / Eine Unity Instanz mit Mosaic Treiber) klar hervorheben. Die dadurch erreichte hohe Flexibilität, die obsoleete Synchronisierung und die gute Performance durch Verteilung der Last auf verschiedene GPUs waren ausschlaggebend.

<sup>1</sup> CVPR - <https://www.cpvrLab.ti.bfh.ch/>

<sup>2</sup> CAVE - [https://de.wikipedia.org/wiki/Cave\\_Automatic\\_Virtual\\_Environment](https://de.wikipedia.org/wiki/Cave_Automatic_Virtual_Environment)

<sup>3</sup> OpenGL - <https://www.opengl.org/>

<sup>4</sup> OpenSceneGraph - <http://www.openscenegraph.org/>

<sup>5</sup> Equalizer - <http://www.equalizergraphics.com/>

<sup>6</sup> Chromium - <http://chromium.sourceforge.net/doc/index.html>

<sup>7</sup> Unity - <https://unity3d.com/>

<sup>8</sup> Nvidia Mosaic - <http://www.nvidia.com/object/nvidia-mosaic-technology.html>

## 1.4 Unity

Die Entwicklungsumgebung erlaubt die Entwicklung von Computerspielen und anderer interaktiver 3D und 2D-Grafik-Anwendungen. Die Umgebung läuft auf den Betriebssystemen Windows und OS X. Zielplattformen sind neben PCs auch Spielkonsolen, mobile Geräte und Webbrowser.

(Quelle: [https://de.wikipedia.org/wiki/Unity\\_\(Spiel-Engine\)](https://de.wikipedia.org/wiki/Unity_(Spiel-Engine)))



Abbildung 1: Unity Editor, Engine

## 1.5 Nvidia Mosaic

Mosaic ist eine Technologie von Nvidia, um mehrere Graphikkarten über den Treiber zu verlinken und auf einem Desktop darzustellen. Windows wird ein einzelner Output vorgegaukelt, auch wenn physisch mehrere GPUs mit multiplen Ausgängen angeschlossen sind.

*„Die NVIDIA® Mosaic™ Mehrbildschirm-Technologie dient zur einfachen Skalierung jeder Anwendung auf mehrere Bildschirme, und das ohne Softwareanpassungen oder Leistungseinbußen. Durch die Mosaic Technologie werden Mehrbildschirm-Konfigurationen vom Betriebssystem als einzelner Bildschirm wahrgenommen.“*

(Quelle: <http://www.nvidia.de/object/nvidia-mosaic-technology-de.html>)

## 1.6 Infrastruktur

Nach Abschluss der Projekt 2 Arbeit wurden basierend auf den gewonnenen Erkenntnissen und den gestellten Anforderungen die Infrastruktur überarbeitet und erweitert. Konkret beinhaltet das folgende Punkte:

- Ersetzen des Servers für das WorldViz Trackingsystem
- Beschaffung und Inbetriebnahme des Unity Servers
- Beschaffung eines Gamepads

Neu gestaltet sich die Infrastruktur wie folgt:

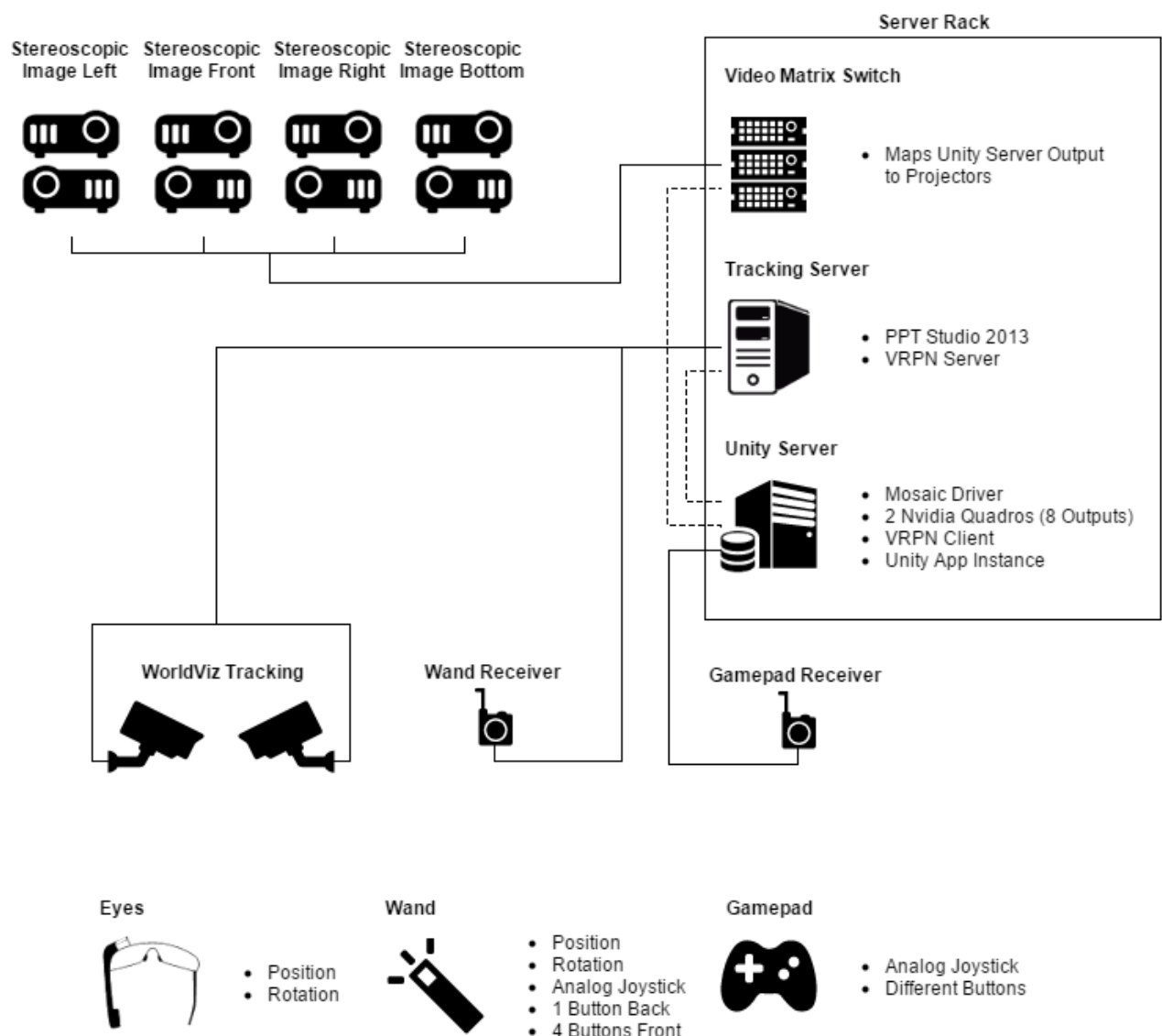


Abbildung 2: Infrastruktur CAVE

### 1.6.1 Inputs und Tracking

Die beiden Tracking Devices (Eyes und Wand<sup>9</sup>) werden durch 10 Infrarotkameras (WorldViz<sup>10</sup> Tracking) geortet. Die Signale werden vom Tracking Server interpretiert und eine genaue Position und Rotation der beiden unabhängig voneinander bewegten Devices werden bestimmt. Auf diesem Server läuft die von WorldViz herausgegebene Software „PPT Studio 2013“, welche Einstellungen des Trackingsystems (Kalibrieren der Kameras, Überprüfen der Signale, Definition der Devices usw.) erlaubt. Zusätzlich wird, direkt im PPT Studio 2013 integriert, ein VRPN<sup>11</sup> Server betrieben, welcher die berechneten Werte der Tracking Devices über einen Socket ins Netz liefert.

Der Wand hat zusätzliche Inputs wie ein analoger Joystick, ein Button auf der Rückseite und 4 Buttons auf der Vorderseite. Diese Inputs werden per Funk an eine Station, den Wand Receiver, geschickt und an den Trackingserver weitergeleitet.

<sup>9</sup> Eyes und Wand - <http://www.worldviz.com/products/ppt/wand-and-eyes>

<sup>10</sup> WorldViz - <http://www.worldviz.com/>

<sup>11</sup> VRPN - <https://en.wikipedia.org/wiki/VRPN>



Abbildung 3: WorldViz Wand

Das Gamepad wird nicht durch das Tracking System geortet, sondern dient als Ersatz für standardmäßige Inputs wie Maus und Tastatur. Das Signal wird direkt an den Unity Server übermittelt.

#### 1.6.2 Rendering

Auf dem Unity Server läuft die Unity Applikation und die über das VPRN Protokoll gelieferten Informationen bezüglich der Tracking Devices werden, mit Hilfe des entwickelten Unity Plugins, interpretiert. Dank des Mosaic Treibers von Nvidia wird das Output-Signal, welches von der Unity Applikation gerendert wird, auf die 8 vorhandenen Grafikkarten Outputs verteilt und an den Video Matrix Switch überliefert.

### 1.6.3 Darstellung im CAVE

Als letzten Schritt empfangen die Projektoren das Signal über den konfigurierten Matrix Switch und stellen eine stereoskopische Ansicht sicher, indem jeweils zwei Projektoren leicht versetzte Bildinformationen erhalten und zusammen eine Leinwand beleuchten. Mittels Polfilter an den Projektoren und einer Polfilterbrille nimmt der Benutzer ein 3D-Bild wahr.

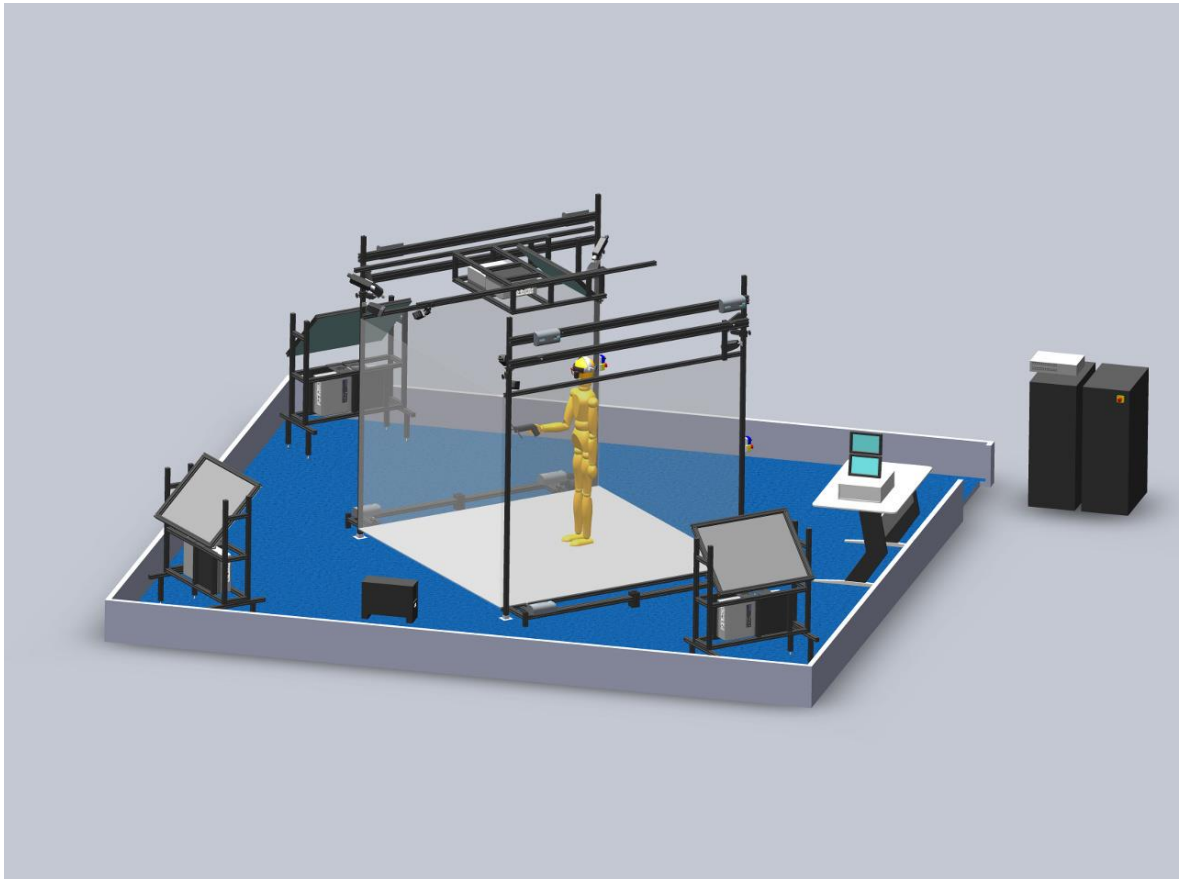


Abbildung 4: CAVE BFH

## 2 Umfang und Ziele der Bachelor Thesis

Folgende vier Schwerpunkte bestimmen den Umfang der gesamten Thesis:

1. Unabhängiges Unity Plugin, welches die Konfiguration für den CAVE ermöglicht. Hier muss besonders auf die Konfiguration des Nvidia-Mosaic Rücksicht genommen werden.
2. Einbindung der bereits installierten Motion-Tracking Lösung von WorldViz ins Unity (über das oben genannte VRPN-Plugin).
3. Implementierung einer/mehreren Unity-Demoapplikationen, welche die Fähigkeiten dieser Installation demonstriert.
4. Erstellen eines Handbuchs sowie ein Tutorial für kommende Unity Anwendungen (der Aufwand für die Portierung in den CAVE sollte extrem einfach und minimal gehalten werden).

Die Zieldefinition ergibt sich direkt aus diesen Anforderungen:

***Es muss möglich sein, mit geringem Aufwand eine gängige Unity-Applikation in kurzer Zeit im CAVE der BFH zu verwenden.***

Die neue und bereits vorhandene Infrastruktur soll benutzerfreundlich und ansprechend in das Projekt eingebunden werden.

### 2.1 Abgrenzungen

#### 2.1.1 Technische Abgrenzungen

Um eine möglichst saubere, wartebare und moderne Applikation anbieten zu können, wird nach Möglichkeit nur auf Unity, respektive C# gesetzt. Low-Level Implementationen in C, C++ (auch FreeGlut) werden keine vorgenommen.

#### 2.1.2 Weitere Abgrenzungen

Als Hardwarekonfiguration des Unityrechners wird die Empfehlung aus dem Projekt 2 umgesetzt / übernommen. Die WorldViz Installation findet, ausgenommen der neuen Servern, wie anhin Verwendung. Es sind höchstens Kalibrierungsarbeiten oder Softwarekonfigurationen vorgesehen.

### 2.2 Voraussetzungen und Ressourcen

Die Voraussetzungen wurden durch das Projekt 2 abgeklärt und bei Bedarf beschafft und installiert. Desweiteren muss der Zutritt zu den Räumlichkeiten der BFH, wo sich die Installation des CAVEs befindet, sichergestellt werden.



### 3 Funktionale Anforderungen

Folgende Tabelle dient als Übersicht, die einzelnen Punkte sind in den Unterkapiteln beschrieben:

|                                       | Priorisierung | Keyfeatures  | Kritische Punkte   |
|---------------------------------------|---------------|--|--|
| Adaption Unity Anwendung für den CAVE | 1             | <ul style="list-style-type: none"> <li>- Automatische Erstellung der zusätzlichen Unitykameras</li> <li>- Rotierung der Unitykameras</li> <li>- Justierung des Frustums und FOV</li> </ul> | <ul style="list-style-type: none"> <li>- Optimale Frustums, welches die genaue Position im CAVE als Unitykamera übernehmen.</li> </ul> |
| VRPN Unterstützung                    | 1             | <ul style="list-style-type: none"> <li>- Implementierung des Protokolls für VR-Eingabegeräte</li> </ul>  | <ul style="list-style-type: none"> <li>- VRPN Server von WorldViz ist eine Blackbox (Gerätenamen)</li> </ul>                           |
| Kompatibilität                        | 2             | <ul style="list-style-type: none"> <li>- Aktuelle Unityversion und zukünftige sollen supportet werden</li> </ul>   | <ul style="list-style-type: none"> <li>- Abhängig von Unity</li> </ul>   |
| Tracking                              | 1             | <ul style="list-style-type: none"> <li>- Erkennung der Positionen und Rotationen von Kopf und WAND im CAVE</li> </ul>  | <ul style="list-style-type: none"> <li>- Abhängig von den vorhandenen Kameras sowie deren Bildqualität</li> </ul>                      |
| Einstellungsmöglichkeiten             | 2             | <ul style="list-style-type: none"> <li>- Alle Freiheitsgrade müssen pro Anwendung einstellbar oder direkt sperrbar sein</li> </ul>   |  |
| Setup, Dokumentation                  | 2             | <ul style="list-style-type: none"> <li>- Plug &amp; Play für Unityanwendungen</li> </ul>   | <ul style="list-style-type: none"> <li>- Spezielle Anwendungen können eventuell nicht durchgängig unterstützt werden</li> </ul>        |
| Demoapplikationen                     | 3             | <ul style="list-style-type: none"> <li>- Siehe Tabelle 2</li> </ul>  |  |
| Unity Output Warping                  | 4             | <ul style="list-style-type: none"> <li>- Das Ausgabebild soll auf die Eigenheiten des CAVEs angepasst werden können</li> </ul>   | <ul style="list-style-type: none"> <li>- Optionales Feature</li> </ul>   |

Tabelle 1: Übersicht Funktionale Anforderungen

#### 3.1 Adaption Unity Anwendung für den CAVE

Beliebige Spiele, Simulationen oder sonstige Anwendungen die mit Unity umgesetzt wurden, sollen so manipuliert werden, dass auf sämtlichen Leinwänden des CAVEs eine stereoskopische Projektion dargestellt wird.

#### 3.2 VRPN Unterstützung

VR-Geräte sollen über das VRPN Protokoll angesprochen werden können und ihre Inputs über das Plugin ins Unity übertragen werden können (WorldViz Wand und Eyes).

#### 3.3 Kompatibilität

Sämtliche, quelloffene Unity Anwendungen ab Version 5.0 sollten mit dem umgesetzten System kompatibel sein. Der Export der Unity Anwendung muss für das spätere Einpflegen in den CAVE für Windows Desktop erfolgen.

Die Schnittstelle zum umgesetzten System kann mit verschiedenen Methoden erfolgen.

- **Dynamic Link Library<sup>12</sup> (.dll)**  
Unabhängig von Managed und Native Plugins, können sämtliche Funktionen über eine kompilierte .dll erfolgen, die ins Projekt integriert werden muss. Diese Methode hätte den Vorteil, dass der Code nicht eingesehen und modifiziert werden kann. Code Completion wird dank der .dll gewährleistet.
- **Asset Store (Packages)**  
Um das Integrieren einer Library zu vereinfachen, kann im Asset Store ein Package angeboten werden, welches direkt an den vorgesehenen Ort kopiert und mit dem Projekt verknüpft wird. Das Package kann unter anderem eine .dll oder offener Code beinhalten.
- **Source Code API<sup>13</sup>**  
Die Schnittstelle kann über offenen Source Code erfolgen. Die entsprechenden Klassen werden ins Unity integriert und können bei Bedarf adaptiert werden. Maximale Flexibilität wird gewährleistet.

Die optimal passende Variante wird durch Prototypen evaluiert. Diese werden bewertet nach folgenden Kriterien:

- **Konfigurierbarkeit**  
Können die nötigen Parameter (CAVE Abmessungen, 6DoF<sup>14</sup>-Einstellungen, und weiteres) gepflegt werden.
- **Plattformunabhängigkeit**  
Werden alle gewünschten Betriebssysteme und Architekturen unterstützt.
- **Skalierbarkeit**  
Können die Features (Tracking, 3D-Stereoskopie) getrennt verwaltet werden.

### 3.4 WorldViz Tracking

#### 3.4.1 Head Tracking

Die Position und Rotation der Hauptkamera ist durch die Unity Anwendung gegeben und kann durch unterschiedliche Inputs (z.B. Maus, Tastatur, Gamepad) erfolgen.

Zusätzlich zu der von der Anwendung definierten Kamerabewegung erfolgt eine leichte Verschiebung und Drehung der Kamera durch das Infrarot Tracking. Diese Translation und Rotation ist aber nur eine minime Veränderung des Kopfes in Relation zur Änderung, die ohnehin von der Applikation gegeben ist.

Entscheidend ist, dass die zwei verschiedenen Inputs, Unity und WorldViz, klar getrennt werden. Sonst kann u.U. eine Situation entstehen, welche nicht definiert ist (Kombinationen der jeweiligen Positionen und Rotationen). Ansonsten wäre eine generische Lösung, die möglichst alle Applikationen abdeckt, unrealistisch.

Die Immersion wird durch diese Methode deutlich gesteigert, weil sich die Oberkörper-, bzw. Kopfbewegung in der virtuellen Welt genau gleich wie in der realen Welt verhält.

Folgende Inputs des Head Tracking Devices werden vom Unity Plugin interpretiert:

- **Position**  
Verschiebt die Kameras um den Betrag, welcher von einem initialen Punkt aus erfolgte. Rein mit dem Head Tracking ist es nicht möglich, die Steuerung des Spiels (welche die Hauptkamera beeinflussen würde) zu übernehmen. Nur dieses Offset wird interpretiert.

<sup>12</sup> DLL - [https://en.wikipedia.org/wiki/Dynamic-link\\_library](https://en.wikipedia.org/wiki/Dynamic-link_library)

<sup>13</sup> API - [https://en.wikipedia.org/wiki/Application\\_programming\\_interface](https://en.wikipedia.org/wiki/Application_programming_interface)

<sup>14</sup> 6DoF - [https://en.wikipedia.org/wiki/Six\\_degrees\\_of\\_freedom](https://en.wikipedia.org/wiki/Six_degrees_of_freedom)

- **Rotation**

Je nach Ausrichtung des Kopfes, bzw. des Head Trackings, dreht sich auch die Kamera im Spiel.

### 3.4.2 Wand Tracking

Ein weiteres Input Device ist der Wand von WorldViz. Das Tracking dieses Gerätes bewirkt die Steuerung der Applikation, die anstelle einer Tastatur, der Maus oder des Gamepads erfolgen kann.

Folgende Inputs des Wand Tracking Devices werden vom Unity Plugin interpretiert und auf die Applikationslogik so weit wie möglich angewandt:

- **Rotation**

Die Rotation des Wands wird als Mausbewegung interpretiert.

- **Joystick**

Diese Eingabe simuliert das Drücken der Pfeiltasten (und gleichzeitig, wie in vielen Spielen üblich, W, A, S, D)

- **Buttons**

Die verschiedenen Buttons des Wands sind auf die meistüblichen Tastatureingaben abgebildet, die in einem Spiel benutzt werden. (Linke Maustaste, rechte Maustaste, Leertaste, Control, usw.) Die Konfiguration des Unity Plugins lässt aber eine neue Zuordnung zu, um der aktuellen Applikation zu entsprechen.

- **Position**

Da die WorldViz Installation über das VRPN Protokoll die Positionen und Rotationen aller erkannten und verfolgbaren Objekte übermittelt, kann alternativ statt des Head Trackings auch die Position des Wands verwendet werden.

### 3.5 Einstellungsmöglichkeiten für 6DoF

Je nach Anwendung muss die Sensibilität der Achsen angepasst werden können, damit die physischen Bewegungen im CAVE stärker oder schwächer interpretiert werden können. Zusätzlich können selektiv Achsen deaktiviert werden.

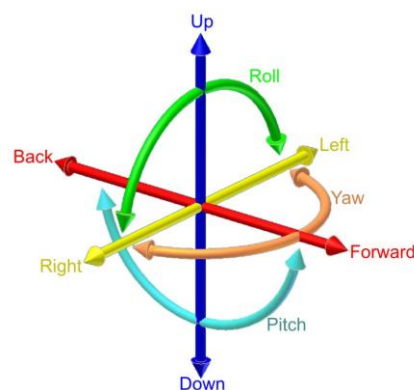


Abbildung 5: 6Dof

### 3.6 Setup

Damit Anwender, die nicht in das Projekt involviert waren, das Unity Plugin problemlos, schnell und einfach mit ihrer eigenen Unity Anwendung benutzen können, wird eine Schritt für Schritt Anleitung und ein Setup erstellt.

### 3.7 Demoapplikation

Um sämtliche umgesetzten Features und die Usability praktisch veranschaulichen zu können, werden Demoapplikationen erstellt, bzw. verwendet.

- **Schiessbude (Priorität 1)**

Diese eigens erstellte Applikation bietet optimale Voraussetzungen, um sämtliche Features des Unity Plugins veranschaulichen zu können.

Das Setting dieses Demospiels ist eine Schiessbude, wie sie auf einem Jahrmarkt anzutreffen ist. Die Galerien mit den abzuschliessenden Zielen verteilen sich jedoch rund um den Spieler. Mit Hilfe des Head Trackings kann sich der Spieler in der gesamten Szenerie umschaun, Bewegungen ausführen und die Objekte aus verschiedenen Perspektiven betrachten. Das Wand-Device steuert das Gewehr, um die Zielobjekte anzuvisieren und abzuschliessen. Die Buttons des Wands werden gebraucht um das Gewehr abzufeuern.

#### **Spiellogik**

Das Spiel startet unmittelbar auf dem Schiessgelände, welches im Wildwest-Setting spielt. Kreisförmig um den Spieler sind verschiedene Marktstände angeordnet, auf denen die abzuschliessenden Objekte erscheinen. Wird ein Zielobjekt mit dem Gewehr getroffen, werden dem Spieler Punkte gutgeschrieben, dessen Summe als GUI-Element in einer Ecke des Frontscreens sichtbar ist. Ein weiteres GUI-Element ist die Zeit, die seit dem Start des Spiels verstrichen ist.

#### **Zielobjekte**

Abzuschliessende Ziele erscheinen an fixierten Positionen, zu einem zufälligen Zeitpunkt und bleiben wiederum eine zufällige Zeitspanne sichtbar. Während dieser Zeit kann mit dem Gewehr das Objekt abgeschossen und Punkte gesammelt werden. Die Meshes der Ziele sind Gummienten und Zielscheiben.

#### **WorldViz Wand**

Der Wand positioniert das virtuelle Gewehr. Ein Fadenkreuz wird entsprechend der Position des Wands angezeigt und dient als Hilfe fürs Abschiessen der Objekte. Das Gewehr richtet sich möglichst genau gleich aus wie der Wand in der realen Welt. Der Button auf der Rückseite feuert das Gewehr ab. Der Joystick hat in dieser Applikation keine Funktion.

#### **WorldViz Eyes**

Damit sich der Spieler in der virtuellen Welt bewegen kann, wird die Position der Eyes aufgezeichnet und in einem definierten Radius ins Spiel übertragen. Die Rotation ist die Ausrichtung des Spielers, also in welche Richtung das Gewehr grundsätzlich zielt. Zu dieser Rotation kommt dann noch die Rotation, welche vom Wand definiert wird, hinzu.

#### **Ziel**

Das Spiel hat kein fixes Ziel, es geht primär darum, möglichst viele Punkte in kurzer Zeit zu sammeln.

- **Model-Viewer (Priorität 2)**

Es wird ein Operationssaal dargestellt, in dem sich der Benutzer im CAVE wie ein Arzt bewegen kann. Als Alternative zum Operationssaal könnten auch Gebäude, Städte und sonstige statische Modelle Verwendung finden. Mit Hilfe des Joysticks auf dem Wand-Device kann sich der Benutzer frei im virtuellen Raum bewegen. Leichte Bewegungen sind mittels Eyes-Device ebenfalls möglich. Spezielles Setting hier ist, dass bspw. ein Körperquerschnitt, MRI- oder Röntgenbild als sekundäre Kamera angezeigt wird, welche über das Plugin konfigurierbar ist. (Fix an einer Seite des CAVEs, fix positioniert in einer Ecke des Front-Screens oder sogar verschiebbar (durch die Wand)).

- **Demoapp Drittpartei (Priorität 3)**

Eine nicht spezifisch auf das Unity Plugin massgeschneiderte Applikation wird ausgewählt, um die Wiederverwendbarkeit und Kompatibilität des erstellten Unity Plugins zu demonstrieren. Möglicherweise können nicht alle Features, die das Plugin bieten würde, vom Spiel interpretiert werden, weil dies von der Spiellogik her nicht möglich ist.

Diese Applikation muss noch evaluiert werden, naheliegender ist das automatisch ausgelieferte Autorenspiel von Unity. Alternativ finden Demo-Apps aus dem Appstore Verwendung.

### 3.8 Features

Alle diese Applikationen stellen verschiedene Anforderungen an das Unity Plugin. Die Apps wurden so gewählt, dass eine möglichst grosse Abdeckung der Features erfolgt. Abhängig vom Projektfortschritt werden möglicherweise nicht alle Demoapps umgesetzt.

|                                  | Schiessbude<br>(Priorität 1) | Demoapp Drittpartei<br>(Priorität 2) | Statische Welt<br>(Priorität 3) |
|----------------------------------|------------------------------|--------------------------------------|---------------------------------|
| Headtracking                     | X                            | X                                    | X                               |
| Wand                             | X                            | ?                                    | X                               |
| Gamepad                          | X                            | ?                                    |                                 |
| Vordefinierte<br>Kamerasteuerung | X                            | X                                    |                                 |
| Vordefinierte<br>Spiellogik      | X                            | X                                    |                                 |
| Sekundäre<br>Kameras             |                              | ?                                    | X                               |

Tabelle 2: Features Unitygames

X: Vorhanden

?: Abhängig der Unity Applikation

### 3.9 Zusätzliche Anforderungen

Aufgrund erkannter Hardwareschwächen im CAVE (unterschiedliche Schärfen der Beamer, Beweglichkeit der Beameraufhängungen und Wärmeausdehnung bei Betrieb) wird folgendes als optionale Anforderung definiert:

#### 3.9.1 Unity-Output Warping

Es soll eine Machbarkeitsstudie durchgeführt werden, ob der Output der Unity-Rendering Pipeline noch mit einem frei justierbaren Filter erweitert werden kann, der Unebenheiten ausgleicht.

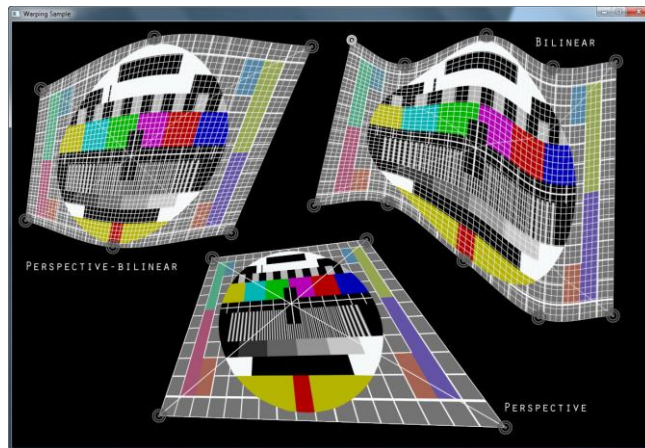


Abbildung 6: Image Wrapping (Quelle: github.com)

## 4 Nicht funktionale Anforderungen

### 4.1 Wiederverwendbarkeit

Damit zukünftige Entwickler effizient eigene Anwendungen in den CAVE einpflegen können, wird viel Wert auf die Wiederverwendbarkeit gelegt.

Anwender aus verschiedenen Bereichen wie Architektur, Autoindustrie, Game Development usw. können ihre Simulationen in den CAVE einpflegen und ausführen.

### 4.2 Ergonomie

Im Rahmen eines kleinen Tutorials wird Schritt für Schritt erklärt, wie die eigene Unity Anwendung für den CAVE aufbereitet werden kann.

### 4.3 Skalierbarkeit sekundäre Kamera

Ein Spiel kann eine sekundäre Kamera verwenden um beispielsweise eine Minimap anzuzeigen. Die Projektion dieser Minimap sollte nicht 1:1 von einem 1-Bildschirm-Setting (fix oben rechts) übernommen werden sondern flexibel sein. Sei dies fix gepinnt an den Center Screen oder dass eine ganze Screen für die Minimap übernommen wird.

Bei sehr spezifischen Kameraeinstellungen können Randcases auftreten, welche nicht adaptiert werden können.



Abbildung 7: Minimap Game

## 5 Testing

### 5.1 System Tests

Während der Prototypingphase werden laufend Tests auf unabhängigen Rechnern sowie im CAVE durchgeführt um sicherzustellen, dass während der Entwicklung mögliche Probleme sofort erkannt werden und Massnahmen ergriffen werden können.

### 5.2 Usability Tests

Abhängig vom Fortschritt der Prototypen werden Tests mit potenziellen Anwendern durchgeführt um die Usability der Lösung abschätzen und optimieren zu können. Sowohl die Inbetriebnahme des CAVEs wie auch die Adaption der eigenen Unity Anwendungen werden berücksichtigt.

### 5.3 Szenarien

Als Testszenarien kommen sämtliche umgesetzten Demoapps zum Einsatz. Somit kann der komplexe Sachverhalt weitgehend geprüft werden. Diese Szenarien erscheinen glaubwürdig, sind komplex und können leicht überprüft werden.

## 6 Administratives

### 6.1 Projektorganisation

Auf eine stark strukturierte Projektorganisation wird bewusst verzichtet. Die Teammitglieder sind gleichberechtigt. Es kann vorkommen, dass verschiedene Teilprojekte und Verantwortungsbereiche den Teammitgliedern zugewiesen werden. Dies bedeutet aber nicht die alleinige Durchführung dieser Tasks.

#### 6.1.1 Projektteam

|                  |  |
|------------------|--|
| Daniel Inversini | <a href="mailto:daniel.inversini@students.bfh.ch">daniel.inversini@students.bfh.ch</a> |
| Julien Villiger  | <a href="mailto:julien.villiger@students.bfh.ch">julien.villiger@students.bfh.ch</a>   |

#### 6.1.2 Betreuer

|                   |  |
|-------------------|--|
| Prof. Urs Künzler | <a href="mailto:urs.kuenzler@bfh.ch">urs.kuenzler@bfh.ch</a> |
|-------------------|--|

#### 6.1.3 Experte

|                   |  |
|-------------------|--|
| Dr. Harald Studer | <a href="mailto:harald.studer@iss-ag.ch">harald.studer@iss-ag.ch</a> |
|-------------------|--|

## 6.2 Projektplan

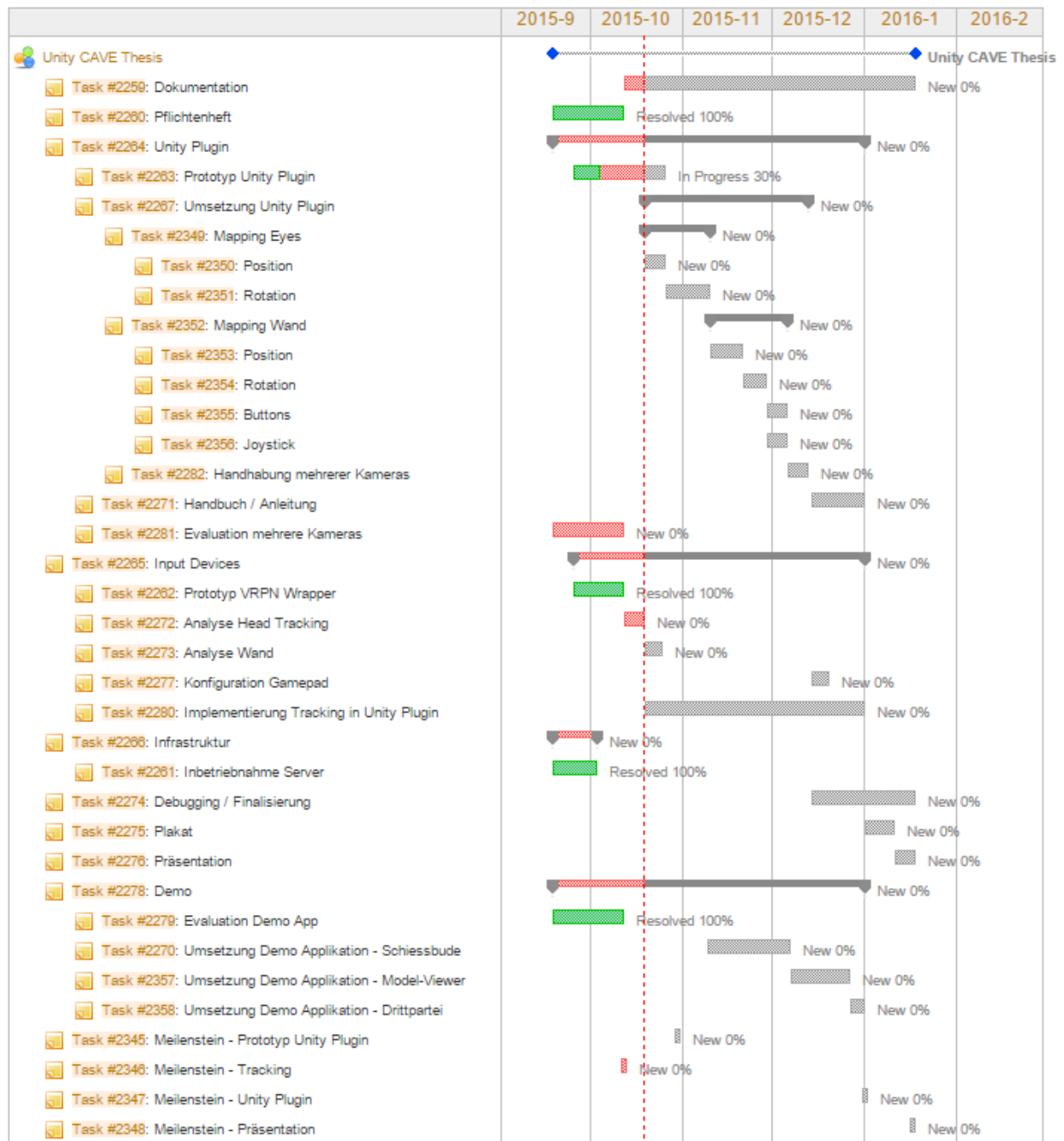


Abbildung 8: Projektplan

<https://pm.ti.bfh.ch/projects/unity-cave-thesis/issues/gantt>

## 6.3 Projektsitzungen

Rund alle zwei Wochen wird ein Projektmeeting des Teams mit Betreuer durchgeführt. Startend ab dem 16. September 2015.

## 6.4 Meilensteine

Folgende Tasks aus dem Projekt wurden als Meilensteine definiert:



#### 6.4.1 Prototyp Unity Plugin

Stichtag 29.10.2015

Eine erste Implementierung des Unity Plugins mit grundlegender Funktionalität wurde umgesetzt.

#### 6.4.2 Tracking

Stichtag 29.10.2015

Die Analysephase des VRPN Protokolls wurde abgeschlossen, damit die Integration des WorldViz Tracking Systems in das Unity Plugin erfolgen kann. Zusätzlich wird die Umsetzung einer Demo-Applikation gestartet.

#### 6.4.3 Unity Plugin / Handbuch / Dokumentation

Stichtag 31.12.2015

Das Unity Plugin wurde fertiggestellt und getestet. Kleinere Anpassungen und die Finalisierung erfolgen noch. Zusätzlich wurde ein Handbuch / Anleitung erstellt, um die Verwendung des Plugins zu vereinfachen. Die Dokumentation ist an dieser Stelle ebenfalls möglichst weit fertiggestellt.

#### 6.4.4 Präsentation

Stichtag 17.01.2016

Sämtliche Dokumente / Arbeiten sind abgeschlossen und eine Präsentation wurde erstellt.

## 7 Versionskontrolle

| Version | Datum      | Beschreibung  | Autor(en)                           |
|---------|------------|---|-------------------------------------|
| 0.1     | 18.09.2015 | Dokument erstellt / Struktur definiert                | Daniel Inversini<br>Julien Villiger |
| 0.2     | 20.09.2015 | Funktionale Anforderungen                             | Julien Villiger                     |
| 0.3     | 01.10.2015 | Infrastruktur / Ausgangslage / allgemeines Updates    | Julien Villiger                     |
| 0.4     | 05.10.2015 | Generelles Update, Review                             | Daniel Inversini                    |
| 1.0     | 06.10.2015 | Generelles Update                                     | Julien Villiger<br>Daniel Inversini |
| 2.0     | 18.10.2015 | Überarbeitung<br>14.10.2015      gemäss      Feedback | Daniel Inversini<br>Julien Villiger |

## 8 Abbildungs- und Tabellenverzeichnis

|   |    |
|---|----|
| Abbildung 1: Unity Editor, Engine .....               | 4  |
| Abbildung 2: Infrastruktur CAVE.....                  | 5  |
| Abbildung 3: WorldViz Wand.....                       | 6  |
| Abbildung 4: CAVE BFH .....                           | 7  |
| Abbildung 5: 6Dof.....                                | 11 |
| Abbildung 6: Image Wrapping (Quelle: github.com)..... | 14 |
| Abbildung 6: Minimap Game .....                       | 14 |
| Abbildung 7: Projektplan .....                        | 16 |
|   |    |
| Tabelle 1: Übersicht Funktionale Anforderungen .....  | 9  |
| Tabelle 1: Features Unitygames .....                  | 13 |

## 9 Glossar

### 6

6Dof \_\_\_\_\_ 2, 9, 11

Six degrees of freedom

Beschreibt die Bewegung eines Körpers im Raum mit Freiheitsgrad 6. Drei unabhängige Richtungen (Translation) und drei unabhängige Achsen (Rotation).

### A

Asset Store \_\_\_\_\_ 9

Unity Asset Store

Der Unity Asset Store ist eine Sammlung von frei verfügbaren wie auch kommerziellen Assets (Plugins) für Unity. Es können Scripts, Models, Texturen, Sound, etc sein. Alles was Unity unterstützt.

### B

BFH \_\_\_\_\_ 3, 7, 8, 18

Berner Fachhochschule BFH  
Die Institution des Studiums.

### C

CAVE \_\_\_\_\_ 1, 2, 3, 5, 7, 8, 9, 11, 13, 14, 18

Cave automatic virtual environment

Bezeichnet den Raum, der verwendet wird um die dreidimensionale Illusionswelt zu erschaffen.

Chromium \_\_\_\_\_ 3

Chromium Rendering System

Bezeichnet ein System, welches fähig ist, interaktives Rendering auf mehreren Clusters durchzuführen.

CPVR \_\_\_\_\_ 3

computer perception and virtual reality

Vertiefungsrichtung des Informatikstudiums an der BFH.

### D

Dynamic Link Library \_\_\_\_\_ 9

Auch oft als DLL verwendet, enthält Programmcode, Daten und Ressourcen für eine Anwendung (EXE).

### E

Equalizer \_\_\_\_\_ 3

Equalizergraphics Parallel Rendering

Middleware-Software, um OpenGL basierende Software auf mehrere Nodes, Graphikkarten und Prozessoren zu verteilen.

Eyes \_\_\_\_\_ 5, 9

#### WorldViz Eyes

Brillengestütztes Trackinggerät, welches mit Infrarot LEDs ausgerüstet ist.

#### F

Frameworks \_\_\_\_\_ 3

Rahmenstruktur, welches dem Programmierer wiederverwendbare Strukturen zur Verfügung stellt.

#### G

Game Engine \_\_\_\_\_ 3

#### Spielengine

Stellt ein spezielles Framework für Computerspiele zur Verfügung. Unterstützt meistens mehrere Plattformen, und verschiedene Funktionalitäten von Spielen (Physik, Graphik, Sound, Netzwerk, KI, etc).

GPU \_\_\_\_\_ 3

#### graphical processing unit

Prozessor, der auf die Berechnung von Graphiken optimiert ist.

#### M

Matrix Switch \_\_\_\_\_ 6, 7

#### DVI Matrix Switch

Stellt ein Routing zur Verfügung von mehreren DVI Eingängen zu mehreren Displays.

MiddleVR \_\_\_\_\_ 3

#### middleVR for Unity

Kommerzielle Applikation, welche die Erstellung von VR Anwendungen unterstützt. Verschiedene Soft- und Hardware werden unterstützt und verbindet sie untereinander.

Minimap \_\_\_\_\_ 14, 19

#### Miniaturn-Map

Stellt einen Überblick über die Spielwelt dar. Abstrahiert und simplifiziert die Details, sodass nur wichtige Akteure (Spieler, Quests, Gelände) dargestellt werden.

Mosaic \_\_\_\_\_ 3

#### Nvidia Mosaic

Die NVIDIA Mosaic Mehrbildschirm Technologie dient zur einfachen Skalierung jeder Anwendung auf mehrere Bildschirme, und das ohne Softwareanpassungen oder Leistungseinbußen.

#### N

Nvidia \_\_\_\_\_ 4

#### Nvidia Corporation

Einer der weltweit grössten Hersteller von Chipsätzen und Graphikprozessoren.

#### O

OpenGL \_\_\_\_\_ 3

#### Open Graphics Library

Plattform- und Programmiersprachen unabhängige Programmierschnittstelle von 2D und 3D Computeranwendungen.

OpenSceneGraph \_\_\_\_\_ 3

Open Source Toolkit für Graphikanwendungen.

OS X \_\_\_\_\_ 4

#### Mac OS X

Kommerzielles Betriebssystem von Apple Computers Inc.

#### P

PPT Studio 2013 \_\_\_\_\_ 5

Software von WorldViz welche die Installation im CAVE der BFH softwareseitig unterstützt (bietet Kalibrierungseinstellungen, VRPN Output und vieles mehr an).

#### U

Unity \_\_\_\_\_ 1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 13, 14, 16, 19

#### Unity Game Engine

Unity ist eine Laufzeit- und Entwicklungsumgebung für Spiele und Simulationen.

#### V

VRPN \_\_\_\_\_ 2, 5, 8, 9, 10, 16

#### Virtual Reality Peripheral Network

Stellt verschiedene Tools (Server, Client, Klassen und Protokolle) zur Verfügung, um ein transparentes Interface zwischen einer Applikation und physikalischen Geräten zu bieten.

## **W**

Wand \_\_\_\_\_ 2, 5, 6, 9, 10, 11, 13, 19

Einhändiger Controller, welcher über Infrarot LEDs verfügt, sowie mehrere Buttons und einen analogen Joystick.

Windows \_\_\_\_\_ 4

Microsoft Windows

Betriebssystem von Microsoft, aktuell in der Version 10.

WorldViz \_\_\_\_\_ 2, 4, 5, 6, 8, 9, 10, 19

Kommerzielle Firma welche komplette Systeme mit Infrarotkameras und Software anbietet.