



ViRGOS

Dokumentation

Hochschule Reutlingen
Masterprojekt VR Lab

Wafa Sadri

SS19-WS19/20

Inhaltsverzeichnis

Einführung

Motivation	3
Grundidee	3

Konzept

Kransystem	4
Haltesystem	4
ViRGOS Lite Konzept	6

Softwaredokumentation

Game Engine	7
Ingame Sound und Communication Sound	7
Doxygen	9

Betriebshandbuch

Voraussetzungen	11
Git Repository	11
Starten des Projektes	12
Gravity Offload System	12
Comms-System	13
Problem- und Fehlerbehebung	14

Quellenverzeichnis

Literaturverzeichnis	15
Weitere Quellen	15

Einführung

ViRGOS - Virtual Reality Gravity Offload System - ist eine Forschungsplattform zur Untersuchung von verschiedenen Schwerkraft- und Lagemodellen im Zusammenspiel mit virtuellen Welten. In diesem Kapitel wird auf die Motivation zu diesem Projekt eingegangen sowie die Grundlagen wissenschaftlich und technisch aufbereitet.

Motivation

Virtual Reality ist längst in vielen professionellen Einsatzbereichen angekommen. Darunter auch in der Luft- und Raumfahrt. Überwiegend wird sie dort als Trainingshilfe verwendet um Piloten und Astronauten auszubilden. Momentan wird Virtual Reality in der Raumfahrt zum Sicherheitstraining, der Roboter-Bedienung (zum Beispiel Canada-Arm2) und dem Bewegen von Objekten im schwerelosen Raum verwendet [1]. Ein bislang nur wenig erkundetes Einsatzgebiet hierbei ist die Verwendung von Virtual Reality für das Extra Vehicular Activity (EVA) Training. Dieses Training findet überwiegend in sogenannten neutralen Auftriebslabors (Neutral Buoyancy Laboratories) statt. Dabei wird Schwerelosigkeit mit Hilfe von Auftrieb im Wasser simuliert [2]. Diese Methode ist jedoch sehr kostspielig, aufwändig und steht oft aus zeitlichen und technischen Gründen nur begrenzt zur Verfügung [3]. Der Einsatz von Virtual Reality (insbesondere von Virtual Reality mit Head Mounted Displays) in neutralen Auftriebslabors ist zwar denkbar, jedoch auf Grund der zuvor genannten Verfügbarkeits- und Kostenprobleme momentan impraktikabel [a]. Aus diesen Gründen entwickelt die NASA ein System um einige dieser Probleme (auch anderer Simulationssysteme) zu beheben [4]. Es handelt sich hierbei um das sogenannte ARGOS (Active Response Gravity Offload System), welches mit Hilfe eines eigens entwickelten Kransystems das Gewicht des Benutzers oder einer Maschine auf verschiedene Schwerkraftmodelle reduzieren kann. Hierbei kommen Elektromotoren zum Einsatz, die eine konstante Zugkraft ausüben können. Dieses Verfahren wird intern als das Active Response System bezeichnet, welches dem ARGOS seinen Namen gibt. Zur Erforschung des Systems in Kombination mit Virtual Reality Technologie, wurde beschlossen das ViRGOS (Virtual Reality Gravity Offload System) an der Hochschule Reutlingen zu entwickeln, dessen Grundidee auf dem ARGOS basiert.

Grundidee

Die Idee von ViRGOS ist es, ein System zu entwickeln, dass ermöglicht verschiedene Schwerkraftmodelle zu simulieren und dabei den Fokus auf den Einsatz von Virtual Reality, insbesondere von Head Mounted Displays, zu legen. Dabei soll eine Plattform entstehen, die verschiedene Möglichkeiten zur Erforschung von Gleichgewicht und Beschleunigung in virtuellen Welten bietet. Dabei sollen mindestens drei verschiedene Schwerkraftmodelle simuliert werden können: Mond, Mars und schwerelos. Weitere denkbare Anwendungs- und Forschungspunkte könnten Tauch- und Flugsimulationen sein. Das ViRGOS soll auf einem Kransystem basieren, dass eine freie, mechanische Bewegung in 2 Achsen und eines Zuges für die 3. Achse ermöglicht. An den Kran soll über ein System aus elastischen Seilen und einem geeigneten Haltesystem ein Benutzer gehängt werden. Diesem Benutzer soll es dann möglich sein durch die Elastizität der Seile und der Übernahme eines Teils des Körpergewichtes durch den Kran eine realistische, schwerkraftreduzierte Bewegung über die Labor-Oberfläche durchzuführen. Dabei trägt er ein Head-Mounted Display, welches ihm eine entsprechende Mars oder Mondlandschaft simuliert.

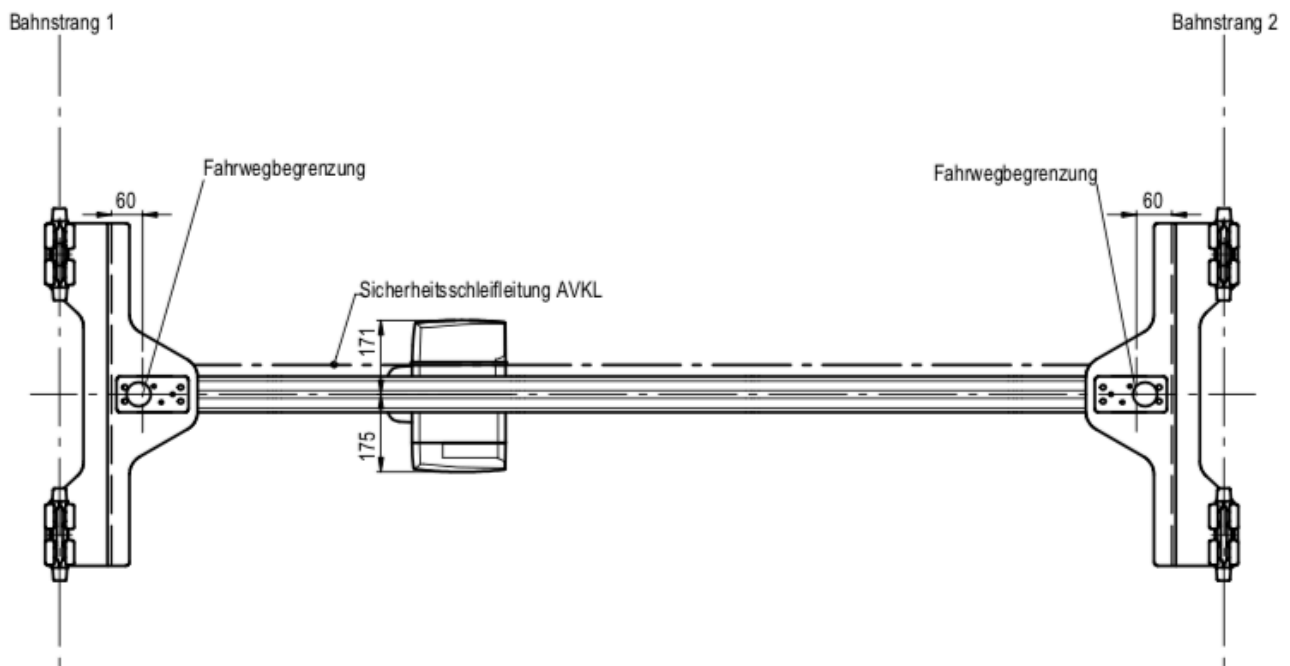


Abb.2: Von ABUS gezeichnetes Kransystem für ViRGOS (siehe [b]). Hier zu sehen: Die Querschiene.

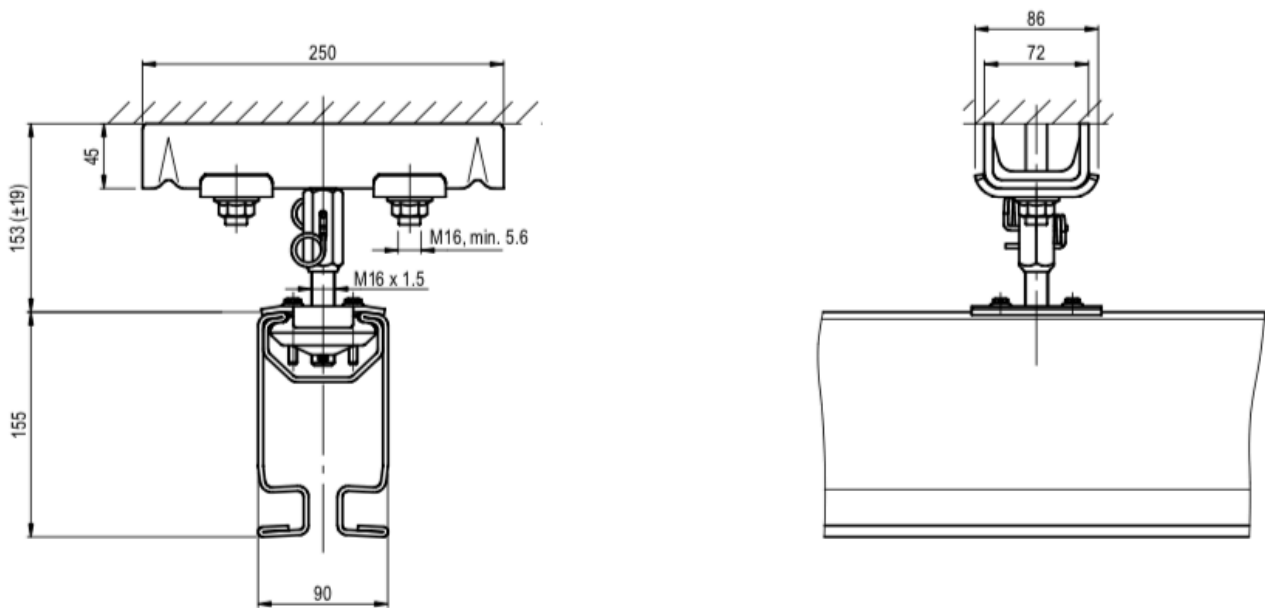


Abb.3: Die Montage der Bahnstränge an der Traverse (siehe [b])

VIRGOS Lite Konzept

Da sich das Kransystem als zu teuer herausgestellt hat, wurde eine entsprechende Lösung entwickelt. Anstelle von sechs Freiheitsgraden an einem freibeweglichen Kran wurde die Traverse direkt als Aufhängung für den User genutzt. Mithilfe eines elastischen Kletterseils und Daisy Chains wird eine entsprechende Zugkraft ausgeübt, die den Benutzer entlastet, und somit ein Mikrogravitationsgefühl erzeugen soll. Da die Aufhängung fest an der Traverse befestigt ist und somit keine freie Bewegung in der horizontalen Ebene gestattet, wurde Virgos Mini in Verbindung mit dem Cyberith Virtualizer entworfen. Die Bewegungen im Virtualizer werden hierbei sogar noch vereinfacht, da die Bodenreibung reduziert wird.

Die Aufhängung besteht aus einer Klettergarnitur, die mit Karabinern und Daisy Chains am elastischen Seilaufbau befestigt ist. Je nach gewählter Länge der Daisy Chains wird die Zugkraft verstärkt oder geschwächt, um somit eine Individualisierung anhand des Gewichts des Benutzers zu ermöglichen. Die entwickelte Lösung erwies sich als deutlich weniger finanziell belastend als das Kransystem und konnte auch entsprechend einfacher umgesetzt werden. Die Interaktionen mit der Welt erfolgen wie auch für das Originalsystem konzipiert über die Controller des Virtual Reality Systems HTC VIVE.



Abb.4: Cyberith Virtualizer



Abb.5: VIRGOS Lite Aufhängung mit Daisy Chains und Klettergurt an der Traverse

Durch das Einbinden des Virtualizers war es auch möglich, auf das haptische Feedback zurückzugreifen, was die Immersion der Benutzer deutlich steigern soll. So kann eine entsprechende Vibration bei Benutzung der Aufzüge aktiviert werden, um das Gefühl einer interaktiven Umgebung zu vermitteln. Die Kosten des Systems belaufen sich auf etwa 250€, was eine substantiell geringere Summe darstellt als das zunächst geplante Kransystem, wessen Anschaffung, Einbau und Abnahme sich auf etwa 8.000€ gerechnet hätten.

Der Kompromiss ist jedoch eine deutlich weniger granular einstellbare Zugkraft, statt einer fließenden Einstellung durch eine Kranwinde erfolgt diese hier über Daisy Chains, welche nur diskrete Schritte in der Zugkraft zulassen. Der Virtualizer selbst erzeugt ebenfalls eine andere Art der Bewegung, die sehr unnatürlich wahrgenommen werden kann, da keine tatsächlichen Schritte, sondern viel mehr „Rutscher“ vollzogen werden müssen, um sich innerhalb des Systems fortzubewegen.

Softwaredokumentation

In diesem Kapitel geht es um die Anforderungen an die Software und welche Software für den Einsatz in ViRGOS ausgewählt wurde.

Game Engine

Als Game Engine wird Unity in der Version 2019.1.0f2 (die aktuellste Version der Unity Engine zum Start der Entwicklung). Unity bietet eine Cross-Plattform-Lösung und eine gute Anbindung an SteamVR, die zur Darstellung des Spiels im Vive Headset (und der Einbindung der Controller) genutzt wird. Des Weiteren erlaubt Unity durch seinen Asset Store ein rapides Prototyping, da viele kompatible Grafikelemente direkt heruntergeladen werden können und so schnell für den Einsatz in Unity bereit stehen.

Ingame Sound und Communication Sound

Zur Steigerung der Immersion wird eine Kommunikationsverbindung vom VR User zum "Commander" hergestellt, wobei es sich bei letzterem um eine Person am Laptop handelt, welcher zusammen mit dem VR User auf einem Teamspeak-Server verbunden ist. Mit Hilfe von Soundfilter-Plugins in Teamspeak ist es möglich, eine Audioqualität ähnlich der des Apollo Programms in den 1960er Jahren zu realisieren. Dabei wird nicht nur ein Rauschen auf die Audiospur gelegt, sondern auch noch entsprechende Audiosignale wie etwa das Piepsen beim Start bzw. Ende einer Audiotransmission. Der Commander wird während dem Betrieb Anweisungen an den VR User schicken, etwa eine Aufforderung zum Ablassen des Kabinendrucks oder zum Öffnen der Türen. Die genauen Wortlaute der Anweisungen sind hierbei frei vom Commander wählbar, solange sie mehr oder weniger den vorgesehenen Ablauf der in VR verfügbaren Aktivitäten wiedergeben.

Verwendete Software:

1. Teamspeak3 Server
2. Teamspeak3 Client für VR User und Commander
3. RadioFX Plugin

Der Teamspeak-Server kann auch auf einem anderen Host installiert werden, entscheidend ist lediglich die Verbindung von VR User und Commander sowie die Nutzung des RadioFX Plugins mit den vorgegebenen Einstellungen. Erfahrungen im VR Labor haben gezeigt, dass es empfehlenswert ist, den Commander außerhalb der Hörweite des VR Users zu platzieren, da somit Rückkopplungen, Crosstalking und andere Beeinträchtigungen der Audioerfahrung ausgeschlossen werden.

Innerhalb von Unity werden Sounddateien abgespielt um die Immersion zu steigern. Die Umsetzung dieser Sounds erfolgt über **AudioSource** Komponenten innerhalb der Szene (siehe Unity-Dokumentation, <https://docs.unity3d.com/ScriptReference/AudioSource.html>).

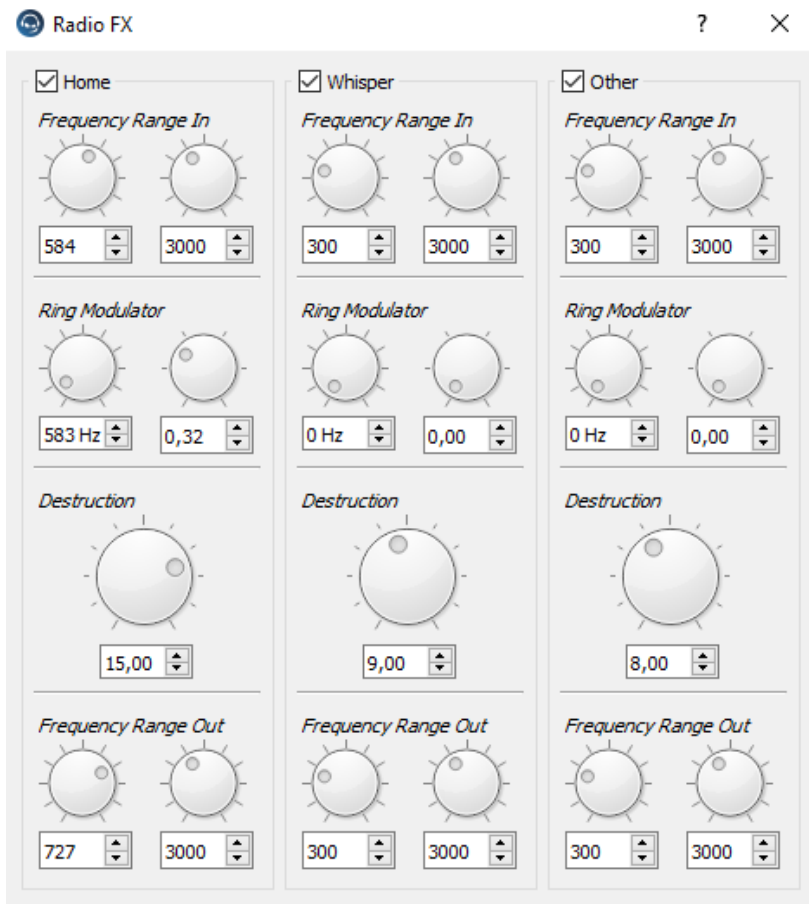


Abb.6: Verwendete Einstellungen von RadioFX

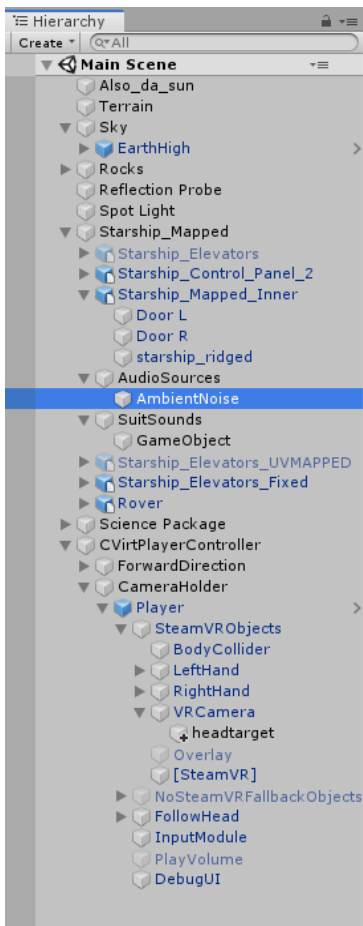


Abb.7: Objekthierarchie von ViRGOS. Hervorgehoben ist das Containerobjekt für AmbientNoises

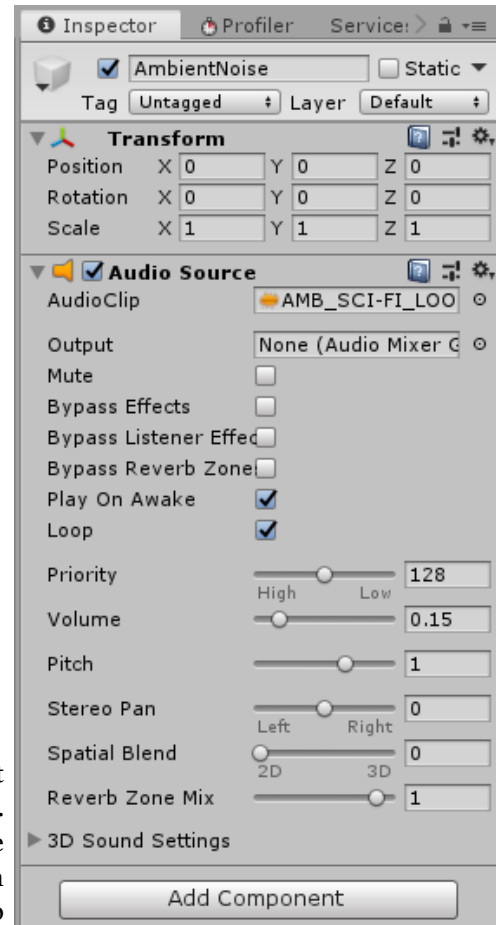


Abb.8: Inspector-Ansicht von Ambient Noise. Sichtbar die Audio Source mit dem dazugehörigen AudioClip

Um die Sounds letztendlich abzuspielen ist eine *AudioListener* Komponente erforderlich, welche dem VR User-Objekt zugewiesen ist (siehe Unity-Dokumentation, <https://docs.unity3d.com/ScriptReference/AudioListener.html>).

Die *AudioSource*-Komponenten selbst werden als Komponenten von *GameObjects* realisiert, welche den *GameObjects* *AudioSources* und *SuitSounds* untergeordnet sind, letztere sind Unterobjekte des Starships. Die Lautstärke der Hintergrundgeräusche (*AmbientNoises*) wird über einen Zeitraum von zehn Sekunden gleichmäßig auf null geregelt, wenn der Druck aus der Kabine abgelassen wird. Umgekehrt wird die Lautstärke langsam von null auf 100% zurückgeregelt, wenn der Druck wieder aufgebaut wird. Hintergrund ist die physikalische Tatsache, dass Schallwellen nicht durch ein Vakuum geleitet werden können. Alle *AudioSources*, die ein Teil von *AmbientNoise* bzw. Kinder von *AudioSources* sind, werden von dieser Regelung betroffen sein. Lediglich *SuitSounds* werden immer abgespielt, da der Raumanzug unter Druck gesetzt und somit in der Lage ist, Schallwellen zu leiten. **Dies ist beim Einfügen neuer Audioelemente zu berücksichtigen!**

Zum Einfügen neuer Sounds ist es lediglich erforderlich, ein entsprechendes *GameObject* mit einer *AudioSource* Komponente zu erzeugen und an das richtige Objekt zu hängen. Die zu abspielende Sounddatei wird als *AudioClip* innerhalb der *AudioSource*-Komponente eingehängt, siehe dazu Abbildung 7. *AmbientNoise* wird generell sofort beim Start und als Endlosschleife abgespielt, da sie Hintergrundgeräusche darstellen sollen. Soll ein Sound nur bei Bedarf abgespielt werden ist das Steuern über ein entsprechendes C#-Skript notwendig, siehe dazu die oben verlinkte Unity-Dokumentation, speziell die Funktion *AudioSource.Play()*.

Doxygen

Die für ViRGOS erstellten C#-Scripts wurden entsprechend dokumentiert und sind im bscw unter dem ViRGOS Ordner im Unterverzeichnis „Doxygen“ zu finden. Doxygen ist ein automatisiertes Tool zum automatischen Generieren von Dokumentationen anhand von JavaDoc oder C#-Summaries. Die Ausgabe erfolgt wahlweise über HTML, XML oder PDF. HTML wurde als Format für die Dokumentation von ViRGOS gewählt da Doxygen mitunter die Vererbungshierarchien visualisiert und Navigationselemente innerhalb der Dokumentation zur Verfügung stellt, darunter auch eine Suchfunktion.

Beim Erstellen neuer Scripts ist darauf zu achten, eine Summary über der entsprechenden Klasse und Methode zu erzeugen. Dazu müssen lediglich drei Kommentar-Slashes in Folge eingegeben werden, Visual Studio erzeugt darauf hin die entsprechenden Kommentar-Header. Eine Summary sollte eine kurze, prägnante Beschreibung einer Methode/Klasse enthalten und die Parameter sowie Rückgabewerte entsprechend beschreiben. Mit Hilfe des <see> Tags kann eine Andere Klasse, Methode oder ein Parameter entsprechend referenziert werden. Sowohl Visual Studio als auch Doxygen werden diese Tags benutzen um in der Dokumentation einen Link auf das angegebene Element zu generieren. Dies soll die Übersichtlichkeit und Nutzererfahrung hinsichtlich der Dokumentation steigern. Beim Generieren via Doxygen ist HTML mit Navigationsleiste als Ausgabe zu wählen. Anschließend die bestehende Doxygen im BSCW löschen und durch die neue ersetzen. Längerfristig ist eine Wiki oder ein Webserver, auf dem die Dokumentation zu finden ist eine nachhaltigere Lösung als das manuelle Einpflegen der Daten ins BSCW.

Der Ordner, den Doxygen zum Generieren verwendet, sollte ViRGOS/Assets sein, allerdings ohne die Auswahl des rekursiven Scans, da sonst alle Scripts im gesamten Projekt miteinbezogen werden.

Provide some information about the project you are documenting

Project name:

Project synopsis:

Project version or id:

Project logo: No Project logo selected.

Specify the directory to scan for source code

Source code directory:

☐ Scan recursively

Abb.9: Metadaten für Doxdoc Generierung

Select the output format(s) to generate

☒ HTML

☐ plain HTML

☒ with navigation panel

☐ prepare for compressed HTML (.chm)

☒ With search function

☐ LaTeX

☒ as intermediate format for hyperlinked PDF

☐ as intermediate format for PDF

☐ as intermediate format for PostScript

☐ Man pages

☐ Rich Text Format (RTF)

☐ XML

Abb.10: Ausgabe für Doxygen

Das „working directory“ wird von Doxygen zum Ablegen temporärer Dateien benötigt und kann frei gewählt werden, es wird jedoch empfohlen ein neues Verzeichnis dafür anzulegen.

Das Output Directory ist das Verzeichnis, welches schlussendlich die generierten Dateien beinhalten soll. Sind alle Parameter eingestellt, wird mit einem Klick auf „Run Doxygen“ das Programm gestartet und die entsprechenden Dateien generiert.

Zum Öffnen der Dokumentation muss lediglich index.html in einem Webbrowser geöffnet werden. Über die Suchleiste kann nach einer Methode oder Klasse gesucht werden und über das Navigationspanel gelangt man in die Klassenübersicht.

File Settings Help

Step 1: Specify the working directory from which doxygen will run

Step 2: Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation

Status: not running

Output produced by doxygen

Abb.11: Working Directory für Doxygen

Betriebshandbuch

In diesem Abschnitt des Dokuments ist ein Betriebshandbuch für ViRGOS gestellt. Hier wird beschrieben, wie ViRGOS betrieben wird, welche Schritte vorgenommen werden müssen, was es zu beachten gibt und welche Schritte bei auftretenden Problemen vorzunehmen sind. Des weiteren werden im Abschnitt zum GOS wichtige Sicherheitshinweise erläutert.

Voraussetzungen

Um das Projekt starten zu können, muss auf dem Zielrechner eine Installation von Unity, bzw. Unity Hub vorhanden sein. Die Version, die für ViRGOS verwendet wurde ist 2019.1.0f2. Falls diese Version nicht zur Verfügung steht, sollte diese installiert werden. Für den Fall, dass eine Installation der o.g. Version nicht möglich ist, sollte ViRGOS auf die aktuellste Version geupdated werden. Dabei gilt zu beachten, dass einige verwendete Funktionen von ViRGOS in der Zwischenzeit aus Unity Deprecated sind und daher möglicherweise nicht mehr funktionieren. In diesem Fall müssen die entsprechenden Methoden neu geschrieben werden.

Ebenso ist zur Verwendung von ViRGOS eine HTC Vive (mit Lighthouse), SteamVR sowie ein Cyberith Virtualizer vorausgesetzt. Die Verwendung des Hubsystems wird empfohlen, ist jedoch aus technischer Sicht optional. Zum Download, bzw. dem Klonen des Repositories ist eine Internetverbindung vorausgesetzt.

Git Repository

ViRGOS wurde über die Softwareversionsverwaltung GitLab entwickelt. In GitLab finden sich daher alle relevanten Daten und Dateien zur Wiederherstellung des Projektes. Um Zugang zum Projektverzeichnis, bzw. dem Git Repo zu erhalten, bitte Alexander Brückner (alexander.brueckner@student.reutlingen-university.de) oder Wafa Sadri (wafa.sadri@icloud.com) kontaktieren. Da ein Teil des Codes in ViRGOS proprietär ist, kann das Repository nicht public gestellt werden. Es gilt zu beachten, dass dies auch in Zukunft gilt!

Im Repo befinden sich mehrere Branches. Zum Zeitpunkt dieses Schreibens befinden sich neben dem Master-Branch auch die Branches *dev*, *dev-ik* und *dev-haptics*. Zur hauptsächlichen Entwicklung von ViRGOS ist der Branch *dev* vorgesehen. Sämtliche anderen Branches sind Feature-Branches und sollten daher nach der Fertigstellung des entsprechenden Features nicht weiter benutzt werden. Zum Zeitpunkt der Verfassung dieses Dokuments, gilt der Branch *dev-haptics* als fertiggestellt.

Vor jedem minor Release sollte der aktuelle Feature-Branch in *dev* merged werden. Dies kann über Merge Requests geschehen. Für fertige Versionen ist ein Merge in den Master-Branch vorgesehen.

Um das Projekt aus dem Git Repo auszuchecken, muss auf dem Zielrechner zunächst ein Projektverzeichnis angelegt werden. Anschließend kann mit...

```
git clone https://gitlab.com/McWaffel/virgos.git
```

...das Repository geklont werden. Die lokale Kopie des Projektes beinhaltet sämtliche Dateien die zur Verwendung mit Unity Version 2019.1.0f2 benötigt werden. Je nach Anwendungsfall ist es notwendig in einen anderen Branch zu wechseln. Für die Entwicklung sollte wie o.g. in einen der Feature-Branches gewechselt werden. Ein stabiler Build sollte immer im Master-Branch erzeugt werden. Um einen Branch zu wechseln kann...

```
git checkout dev
```

...verwendet werden.

Starten des Projektes

Das Projekt kann in Unity Hub importiert und geöffnet werden. Zum starten des Projektes kann einfach der Play-Button in Unity verwendet werden.

Gravity Offload System

Bei der Verwendung des GOS ist große Vorsicht geboten. Dies gilt insbesondere bei erstmaligen Usern. Es wird empfohlen den Gurt nicht straff zu ziehen, da durch das Aufhängen der Masseschwerpunkt sonst noch weiter nach oben geschoben wird, was wiederum für ein verstärktes nach-vorne-Fallen sorgt. Stattdessen den Gurt locker tragen.

Sicherheitshinweis: Der Gurt ist so zu tragen, dass die dicke Seite der Aufhängung hinten ist! Der Gurt ist dann falsch herum, wenn sich die dünne, flache Seite hinten befindet! In der Regel sollte sich an der Vorderseite ein nicht-genutzter Karabiner befinden.

Zur Aufhängung wird empfohlen, den Proband zunächst den Gurt anziehen zu lassen und ihm dann die Schuhüberzüge für den Virtualizer zu geben. Anschließend soll der Proband in den Virtualizer einsteigen. Erst dann sollte dem Proband eine Steigmöglichkeit gereicht werden, um auf die Höhe des Karabiners zur Einhängung zu kommen.

Sicherheitshinweis: Es wird davon abgeraten, das Seil zum Probanden herunter zu ziehen, da hier enorme Spannung auf dem Seil herrscht und bei einem Abrutschen schwerste Verletzungen die Folge sein können!

Ist der Proband auf der richtigen Höhe, kann der Karabiner, welcher am Seil hängt, hinten an die dicke Aufhängeschlaufe des Gurts befestigt werden. Beim Abstieg empfiehlt es sich an den Säulen des Virtualizer halt zu suchen. Dies gilt insbesondere für Erstbenutzer, da das Gefühl leichter zu sein zunächst ungewöhnlich erscheint. Reicht der Proband nicht auf den Boden, oder kann er gerade noch die Zehenspitzen auf den Boden bringen, müssen die Daisychains am Seil justiert werden. Dafür muss der Proband zunächst wieder abgehängt werden. Anschließend können am oberen Karabiner, zwischen den Daisychains und dem Bungie die Schlaufen verlängert/verkürzt werden.

Sicherheitshinweis: Niemals das GOS ohne die Daisychains betreiben und auf gar keinen Fall und niemals die Daisychains ohne Zwischenstufe einhängen, einzeln einhängen oder ohne Doppelschlaufe einhängen! Sollte nämlich eine der Chain-Links aufreißen, wäre der Proband komplett ohne Halt und würde auf den Virtualizer stürzen. Verletzungen und Beschädigungen der Hardware sind zur Folge. Unter keinen Umständen darf die Anordnung der Karabiner am Seil geändert werden! Schock-Loads auf die Daisychains sind zu vermeiden. Weitere Informationen zu den Gefahren von Daisychains: https://eu.blackdiamondequipment.com/de_DE/qc-lab-daisy-chain-dangers-en-glbl.html. Die nachfolgenden Grafiken soll die korrekte Anwendung veranschaulichen:

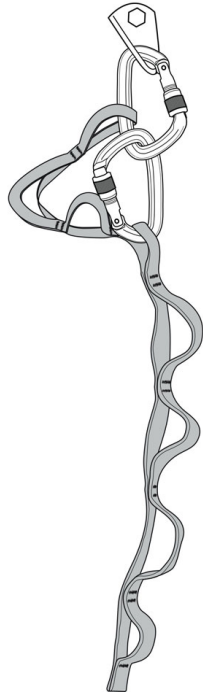


Abb.12: Korrekte Anwendung von Daisy Chains [c]

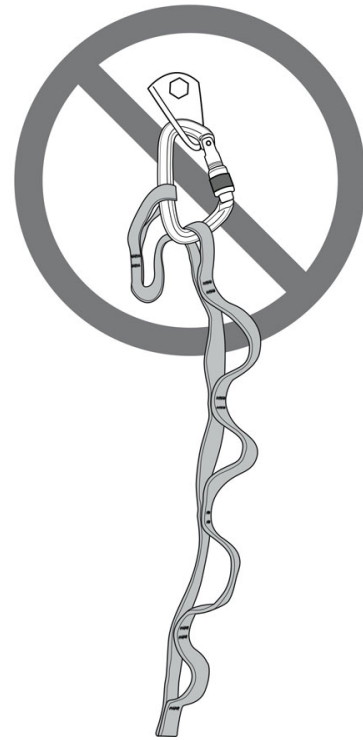


Abb.13: Falsche Anwendung von Daisy Chains [c]

Comms-System

Das Comms-System (oder Communications-System) ist eine optionale Erweiterung für ViRGOS, die eine immersionsstärkende Kommunikation zwischen Proband und Entwickler ermöglichen soll. Dafür wird Teamspeak in Zusammenhang mit einem Voice-Plugin verwendet. Zur Installation und Verwendung von Teamspeak und dem Plugin, siehe Kapitel *Ingame Sound und Communication-Sound*.

Problem- und Fehlerbehebung

Problem/Fehler	Lösung(en)
nach dem Start des Projekts wird eine Exception geworfen	Generelle Lösung: Auf Stable Branch (Master) wechseln Technische Lösung: Prüfen ob irgendwelche Prefabs nicht gesetzt sind, die Version von Unity korrekt ist
Headset/Controller werden nicht getracked	<ul style="list-style-type: none"> - Prüfen ob die Lighthouse-Basestations eingeschaltet sind - Prüfen ob Controller aufgeladen sind - Kabel des Headsets prüfen
Headset tracking setzt manchmal aus	- Proband darauf hinweisen, dass er das Headset nicht anfassen soll während der Simulation
Laufrichtung des Virtualizers ist inkorrekt/funktioniert nicht	<ul style="list-style-type: none"> - Programm anhalten, prüfen ob der Virtualizer nach vorne ausgerichtet ist beim Start des Programms, dann Programm wieder starten - Virtualizer über die Kalibrierungssoftware kalibrieren (siehe dazu Hersteller-Doku) - Prüfen ob alle Datenkabel angeschlossen sind
Virtualizer hat kein haptisches Feedback	<ul style="list-style-type: none"> - Prüfen ob alle drei Kabel (inkl. Stromkabel) angeschlossen sind - Prüfen ob die Steckdose Strom liefert - Kalibrierungssoftware verwenden und Haptic-Test machen um zu prüfen ob es an der Software oder Hardware liegt
Proband reicht nicht auf den Boden im Hubsystem (Gravity Offload System)	- Daisy Chain herabsetzen (siehe dazu das Kapitel „Gravity Offload System“). Sicherheitshinweise unbedingt beachten!
VR Körper nicht an der gleichen Stelle wie Benutzer	- Room Setup von SteamVR neu erstellen
VR Hände versetzt zu echten Händen des Benutzers	- SteamVR-Objekte aus Projekt entfernen, neu importieren

Quellenverzeichnis

Literaturverzeichnis

- [1] S. Aukstakalnis: Practical Augmented Reality - A Guide to the technologies, applications, and human factors for AR and VR; 2017; Addison-Wesley; ISBN 978-0-13-409423-6
- [2] J. Jairala, R. Durkin, Z. Ney, S. Parazynski: EVA Development and Verification Testing at NASA's Neutral Buoyancy Laboratory; 2012; NASA; Houston, TX; 20120006539
- [3] J. R. Braden, D. L. Akin: Development and Testing of a Space Suit Analogue for Neutral Buoyancy EVA Research; 2002; SAE Technical Paper; University of Maryland; 2002-01-2364
- [4] L.K. Dungan, P.S. Valle, D.R. Bankieris, A.P. Lieberman, L. Redden, C. Shy: Active Response Gravity Offload And Method; 2015; NASA, Washington D.C.; US Patent No.: US 9,194,977 B1

Weitere Quellen

- [a] W. Sadri: Visuell unterstützende Trainingshilfen für die bemannte Raumfahrt in neutralen Auftriebslabors; 2019; Hochschule Reutlingen; Fakultät Informatik; *Unveröffentlicht*
- [b] ABUS Kransysteme GmbH Lageplan Angebot 1556509/1; Lageplan des ViRGOS Kransystems im Auftrag entstanden.
- [c] Black Diamond Equipment: Daisy Chain Dangers; https://eu.blackdiamondequipment.com/de_DE/qc-lab-daisy-chain-dangers-en-glbl.html