|  |  |  |
| --- | --- | --- |
| C:\_Hochschule Heilbronn\6_AG MCI\4_Organisation\0_Corporate Design\1_Logo\UniTyLab_Logo.jpg |  | D:\Desktop\HHN Original (Vektor).emf |

Bachelor

Thesis

**Konzeption und Evaluation von Benutzerkonditionierung**

**in Virtual Reality**

|  |  |
| --- | --- |
| **Autor** | Robert Zlomke |
| **Studiengang** | Angewandte Informatik  Hochschule Heilbronn |
| **Matrikel-Nr.** | 193045 |
| **Abgabe** | 15.04.2021 |
| **Referent** | Prof. Dr.-Ing. Gerrit Meixner |
| **Korreferent:** | Philip Schäfer |

* 1. Danksagung
  2. Abstract

Virtuelle Umgebungen zu Fuß zu erkunden ist die realistischste und natürlichste Schnittstelle. Zeitgleich ist dies auch technisch und logistisch am anspruchsvollsten.

Allgemein wurden in bisherigen Arbeiten sich auf die Fortbewegung in übereinstimmenden oder leeren Umgebungen konzentriert. Weniger wurde darauf geachtet, wie sich eine Änderung der Nichtübereinstimmung zwischen der physischen Umgebung und ihrer virtuellen Darstellung auf die Benutzer auswirkt.

**Schlagwörter**: Wissenschaftliche Arbeit, Dokumentvorlage, Hochschule

## Inhaltsverzeichnis

[I. Danksagung 2](#_Toc462041536)

[II. Abstract 3](#_Toc462041537)

[1 Inhaltsverzeichnis 4](#_Toc462041538)

[2 Einleitung 5](#_Toc462041539)

[2.1 Unterpunkt 1 5](#_Toc462041540)

[2.2 Unterpunkt 2 5](#_Toc462041541)

[3 Stand der Technik 6](#_Toc462041542)

[3.1 Unterpunkt 1 6](#_Toc462041543)

[3.2 Unterpunkt 2 6](#_Toc462041544)

[4 Problemstellung, Zielsetzung und Vorgehensweise 7](#_Toc462041545)

[4.1 Problemstellung 7](#_Toc462041546)

[4.2 Zielsetzung 7](#_Toc462041547)

[4.3 Vorgehensweise 7](#_Toc462041548)

[5 Hauptkapitel 8](#_Toc462041549)

[5.1 Unterpunkt 1 8](#_Toc462041550)

[6 Zusammenfassung, Bewertung und Ausblick 9](#_Toc462041551)

[7 Anhang 10](#_Toc462041552)

[7.1 Abkürzungsverzeichnis 10](#_Toc462041553)

[7.2 Abbildungsverzeichnis 11](#_Toc462041554)

[7.3 Literaturverzeichnis 12](#_Toc462041555)

[7.4 Eidesstattliche Erklärung 13](#_Toc462041556)

## Einleitung

### Unterpunkt 1

### Unterpunkt 2

## Stand der Technik

### Unterpunkt 1

### Unterpunkt 2

## Problemstellung, Zielsetzung und Vorgehensweise

### Problemstellung

Virtuelle Umgebungen zu Fuß zu erkunden ist die realistischste und natürlichste Schnittstelle. Zeitgleich ist dies auch technisch und logistisch am anspruchsvollsten.

Allgemein wurden in bisherigen Arbeiten sich auf die Fortbewegung in übereinstimmenden oder leeren Umgebungen konzentriert. Weniger wurde darauf geachtet, wie sich eine Änderung der Nichtübereinstimmung zwischen der physischen Umgebung und ihrer virtuellen Darstellung auf die Benutzer auswirkt

### Zielsetzung

* Konzipieren eines Szenarios, in dem das Bewegungsverhalten der Nutzer über festgelegte Reize in der VR beeinflussen zu können
* Evaluation, um die benötigten Daten zu sammeln und auszuwerten

### Vorgehensweise

* Szenario wird nach der Definition der Konditionierungsreize entwickelt
* Definition der Reize zur Konditionierung der Benutzer in VR (Malus)
* Schnittstelle zur Oculus Quest implementieren
* Auswahl der Erhebungsmethoden und nötigen Daten
* Evaluierung

### Technologien

Für die bestmögliche Umsetzung des Projekts werden verschiedene Technologien benutzt, die im Nachfolgenden erläutert werden.

#### Unity 3D

Unity 3D ist eine Laufzeit- und für Spiele und andere 3D und anderer interaktive 3D-Grafik-Anwendungen. Ebenso das meistgenutzte Tool im Bereich VR-Entwicklung. In Szenen werden die 3D Umgebungen erstellt. Diese besteht zum größten Teil aus Objekten, Materialien bzw. Texturen, Animationen, Audio und Lichtquellen. So entsteht eine virtuelle Welt. Der Android Support ist sehr umfangreich, was der Entwicklung mit der Oculus Quest zugutekommt. [1]

#### Oculus Quest

Bei der Oculus Quest handelt es sich um eine mobile VR Brille. Sie kann also genutzt werden, ohne zusätzlich einen leistungsstarken Computer oder Konsole zu benötigen. Ebenso werden keine externen Sensoren oder Kameras für das Tracking gebraucht.

Die Auflösung des Displays ist 1.600x1.440 Pixel pro Auge mit einer Wiederholrate von 72Hz. Das Tracking passiert über die Brille selbst. Auf dieser befinden sich an der Vorderseite 4 Kameras, die mit Weitwinkel arbeiten. Mit 6DoF, in Englisch Six Degrees of Freedom, ist die Nutzung aller sechs Freiheitsgrade möglich. Somit stehen viele Steuermöglichkeiten zur Verfügung und eine Bewegung im virtuellen Raum gegeben. Dafür elementar ist die Bedienung mit den zwei dazugehörigen Controllern (siehe Abb. 1: Oculus Quest mit Controller). [2]



Abb. 1: Oculus Quest mit Controller[[1]](#footnote-1)

#### Visual Studio

Visual Studio 2019 ist eine Entwicklungsumgebung mit vielen Tools und Dienste für Entwickler. Die eingesetzte Programmiersprache ist hierbei C#. Die Software eigenen Refactorings vereinfachen die Organisation des geschrieben Codes durch vorgeschlagene Aktionen. IntelliCode von Visual Studio gibt dem Entwickler Unterstützung mit dem Einsatz von künstlicher Intelligenz. Beinhaltend sind unter anderem die Bereitstellung von Codevervollständigung und die Suche nach Codeproblemen. Mit der integrierten Codebereinigung ist es möglich mit einem Klick Warnungen und Vorschläge zu ermitteln und zu beheben.

Mit Debugging kann präzise nach Laufzeitfehlern gesucht werden. [3]

#### Programmiersprache C#

C# ist eine objektorientierte und moderne Programmiersprache. Hier werden sogenannte Objekte zur Kapselung von Funktionalität und Daten genutzt. Dies wird erweitert durch die Möglichkeit der Vererbung und Polymorphie. [4] [5]

#### Git Versionskontrolle

Git ist eine kostenlose Open Source Software. Sie dient bei der Softwareentwicklung zur verteilten Versionskontrolle. Die Nutzung wird heutzutage in Unternehmen und von privaten Entwicklern genutzt. Durch die Plattformunabhängigkeit lässt sich Git in fast jeder Entwicklungsumgebung integrieren und nutzen. Die erste Version des Tools wurde im Jahre 2005 veröffentlicht.

Der Nutzen einer Versionskontroller besteht darin, Änderungen verteilt dem Softwareprojekt hinzufügen zu können. Diese werden zusätzlich protokolliert und sind somit nachvollziehbar. Es kann zu einem späteren Zeitpunkt auch auf frühere Versionen zugegriffen werden.

Die Verwaltung geschieht über dezentrale Git-Repositories, im Gegensatz zu anderer Software zur Versionskontrolle, welche mit einer zentralen Datenbank arbeiten. Jeder Entwickler, der an einem Projekt mit Git arbeitet, besitzt eine lokale Kopie des Repositories. Diese wird durch Programmieren weiterentwickelt und die Änderungen mit den anderen Entwicklern des Projekts geteilt.

Ein weiterer großer Bereich ist das Aufteilen und Zusammenführen von verschiedenen Entwicklungsständen. Hierbei kann man ein Softwareprojekt in Unterverzweigungen, den Branchen aufteilen. Dies wird vor allem genutzt, um parallel zum Hauptzweig an verschiedenen Features zu arbeiten. Zum Schluss werden die Branches wieder mit dem Hauptzweig zusammengeführt.

Es gibt viele Webanwendungen, welche Git nutzen und weitere Funktionen hinzufügen. Die Verwaltung von Softwareprojekten wird damit noch übersichtlicher, zugänglicher, transparenter und für andere Entwickler online verfügbar. Der Entwicklungsfortschritt kann grafisch und in Statistiken dargestellt werden. Auch ein eigenes Wiki über das Projekt kann für die Community erstellt und öffentlich zur Verfügung gestellt werden. Die Webseite GitHub ist ein Beispiel für solch eine Plattform. [6]

## Umsetzung

### Einbinden der Oculus Quest in Unity 3D

Um selbst entwickelte grafische Anwendung bzw. Spiele in Unity 3D ausführen oder testen zu können, muss eine aktive Verbindung von der Oculus Quest zum Computer und der Entwicklungsumgebung hergestellt werden.

Um die Verbindung zum PC erfolgreich herstellen zu können, wird die Software Oculus Link benötigt. Diese wird vom Unternehmen selbst online angeboten. Erforderlich ist ebenso das Einrichten der Oculus Quest Brille mit einem Facebook Account.

In Unity 3D ist es nötig, den Oculus Support einzurichten. Dazu gibt es aktuell noch zwei Möglichkeiten, welche ihre Vor- und Nachteile bieten. Das veraltete XR-Plugin Tool ist schon standardmäßig integriert und bietet dazu die Möglichkeit, OpenVR zu nutzen. Dies wird benötigt, wenn weitere VR-Brillen in der Entwicklungsumgebung genutzt werden sollen. Ein Beispiel ist die HTCVive. Da die Umsetzung einzig mit der Oculus Quest durchgeführt wird, fiel die Entscheidung auf das neue XR-Management. Dies befindet sich in ständiger Weiterentwicklung, somit wird in naher Zukunft ebenso OpenVR unterstützt werden. Dies könnte wichtig sein, falls das Projekt weiterentwickelt werden soll.

Ebenso benötigt wird das Asset Oculus Integration, welches sich über den integrierten Asset Store finden und importieren lässt. Darin befinden sich einige Prefabs und Skripte, die benötigt werden, um grundlegende Interaktionen wie das Umschauen und das Gehen in der Virtuellen Welt ermöglichen. Dazu muss das Prefab OVRPlayerController vom Assets Order per Drag and Drop in die Hierarchie der Szene gezogen werden. Zusätzlich wird das Skript CharacterCameraContraint benötigt, welches noch hinzugefügt werden muss.

Ist die Oculus Quest erfolgreich mit Oculus Link verbunden, kann die VR-Brille direkt in der Unity Entwicklungsumgebung über den Play Mode ausgeführt werden. Dies wird vor allem in der Entwicklung benötigt. Für die spätere Evaluation läuft die Anwendung jedoch Standalone, da ohne Kabel und angeschlossenen Computer ein freieres Bewegen der Probanden möglich ist. Dazu werden in den Grafikeinstellungen einige Änderungen vorgenommen und die Plattform wird zu Android gewechselt. Ebenso muss der Account als Developer freigeschalten werden und die Oculus Quest im Entwicklermodus laufen.

### Entwicklung des Malus

Dieser Teil wurde zunächst separat zum Hauptprojekt entwickelt und nach der Fertigstellung integriert. Ziel ist es, einen visuellen und akustischen Malus auszulösen, sobald der Proband vom Weg abkommt. Visuell sichtbar sein soll ein rotes, pulsierendes Blinken im kompletten Sichtfeld. Ebenso soll neben dem visuellen Reiz ein akustisches Piepen ertönen. Die Frequenz beide Stimuli ist zeitgleich und wird bei steigender Dauer schneller.

Für die Umsetzung in Unity 3D wurde das UI Objekt Canvas implementiert. Dies ist dazu da um Inhalte wie z.B. Bilder, Buttons für ein Menü oder Text anzeigen zu lassen. Eine Besonderheit in der Entwicklung einer VR-Anwendung im Vergleich zu einer normalen 3D Umgebung ist, dass das Canvas mit dem Kamera Objekt des OVRPlayerController verknüpft werden muss. Sonst besteht das Problem der fehlenden Anzeige in der VR-Brille.

Unter dem Canvas wurde ein Image Objekt erstellt und die Größe angepasst. Die Farbe wurde Rot wurde als Hintergrundfarbe eingestellt und der Alpha-Wert zunächst auf 0, sodass beim Start der Anwendung die Farbe erstmal nicht sichtbar ist. Für das Erzeugen des Impulses wurde eine Animation erstellt. Diese nimmt den Alpha-Wert des Images und verändert diesen, um so ein Impuls zu erzeugen. Ebenso wurde eine AudioSource Komponente dem Image Objekt hinzugefügt, die dazu benutzt wird, um die Audiodatei einbinden zu können. Diese wurde zuvor per Drag and Drop in die Assets des Projekts gelegt und dient neben dem visuellen zusätzlich noch einen akustischen Reiz. Um das Audio zeitgleich mit dem visuellen Impuls abspielen zu lassen, wurde ein Event PlayAudioSource in der Animation erstellt. Dazu wurde zusätzlich ein Skript dem Image Objekt hinzugefügt, welches das Event implementiert, die AudioSource Komponente aufruft und das Audio abspielt.

Um die Animation ausführen zu können, wird ein Animator benötigt. Dieser bindet die Erstellen Animationen ein und ist später die Schnittstelle, um die Animationen per Skript ausrufen und starten zu können.

Nach erfolgreichem Testen wurde der Canvas als Prefab, der Animator mit den entsprechenden Animationen und der AudioSource in das Hauptprojekt integriert. Der Aufruf des Malus wurde zum späteren Zeitpunkt bei der Entwicklung der Szenarien implementiert.

### Entwicklung der Szenarien

### Datenerfassung

Für die Evaluation wurden zwei verschiedene Arten von Daten vorgesehen. Quantitative Daten, die zur Laufzeit der Anwendung entstehen, und qualitative Daten durch die spätere Befragung der Probanden mit Hilfe eines Online Fragebogens.

Die Daten, die automatisch von Unity während der Laufzeit der Szenarien getracked werden sind:

* welches Szenario aktiv war
* Zeit, die der Proband braucht, um den kompletten Weg bis zum Ziel zu laufen
* die Anzahl der Malus, die aktiviert wurde
* die Dauer pro Malus, als wie lange der Proband gebraucht hat, um wieder auf den richtigen Weg zurück zu kommen

Dazu wurde das Skript DataRecorder erstellt und an das OVRPlayerController Objekt hinzugefügt. In diesem Skript wurden mit C# Methoden erstellt, um die erforderlichen Daten, wie zuvor erläutert, aufzuzeichnen und zu speichern. Die Methoden werden durch das Player Skript aufgerufen.

Das Skript CSVManager ist für den Export der aufgezeichneten Daten in eine CSV Datei zuständig. Dies ist für das Auslesen und weitere Auswertung der Daten entscheidend. Die Datei Report.csv wird lokal im Assets Ordner Report erstellt bzw. abgelegt. Für den Fall, das Daten erhoben und abgespeichert werden sollen, die CSV Datei aber noch nicht existiert, wird diese und der dazugehörige Ordner automatisch erstellt. Dazu werden standardmäßig folgende Header in gleicher Reihenfolge erstellt:

* Szenario
* Gebrauchte Zeit
* Malus aktiviert

Sobald ein Malus öfter als einmal aktiviert wurde, werden die dazugehörigen Header für die Dauer der einzelnen Malus dynamisch hinzugefügt. Ein Beispiel der CSV Datei mit gefüllten Daten zeigt sich in Abb. 2: CSV Datei mit Beispieldaten.



Abb. 2: CSV Datei mit Beispieldaten

Es wurde ein Online Fragebogen erstellt, um zusätzlich noch qualitative Daten zu erheben. Für die Umsetzung wurde EvaSys verwendet.

## Zusammenfassung, Bewertung und Ausblick

## Anhang

### Abkürzungsverzeichnis

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### Abbildungsverzeichnis

[Abb. 1: Oculus Quest mit Controller 8](#_Toc65338790)

[Abb. 2: CSV Datei mit Beispieldaten 12](#_Toc65338791)

### Literaturverzeichnis

|  |  |
| --- | --- |
| [1] | [Online]. Available: https://unity.com/de. [Zugriff am 23 02 2021]. |
| [2] | D. Ziesecke, „Alle Infos zur Oculus Quest – technische Daten, Preis und Laufzeit,“ 03 03 2019. [Online]. Available: https://vr-legion.de/news/alle-infos-zur-oculus-quest-technische-daten-preis-und-laufzeit/. [Zugriff am 23 02 2021]. |
| [3] | TerryGLee, „Neues in Visual Studio 2019,“ 10 11 2020. [Online]. Available: https://docs.microsoft.com/de-de/visualstudio/ide/whats-new-visual-studio-2019?view=vs-2019. [Zugriff am 22 02 2021]. |
| [4] | B. Wagner, „Überblick über C#,“ 28 01 2021. [Online]. Available: https://docs.microsoft.com/de-de/dotnet/csharp/tour-of-csharp/. [Zugriff am 23 02 2021]. |
| [5] | O. B. Dr. Veikko Krypczyk, „Einführung in die Programmierung: Objekte, nichts als Objekte,“ 19 04 2016. [Online]. Available: https://entwickler.de/online/development/einfuehrung-programmierung-objektorientierte-programmentwicklung-197372.html. [Zugriff am 24 02 2021]. |
| [6] | S. Augsten, „Definition „Git SCM“ - Was ist Git?,“ 27 08 2019. [Online]. Available: https://www.dev-insider.de/was-ist-git-a-850847/. [Zugriff am 24 03 2021]. |
| [7] | C. Spinger, „Oculus Quest im Test: Virtual Reality für die Masse,“ 30 04 2019. [Online]. Available: https://www.vrnerds.de/oculus-quest-im-test-virtual-reality-fuer-die-masse/. [Zugriff am 27 02 2021]. |

### Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Heilbronn, 19. September 2016 Vorname Name

1. https://www.vrnerds.de/wp-content/uploads/2019/04/Oculus-Quest-2-1.jpg [7] [↑](#footnote-ref-1)