

Hochschule der Medien Stuttgart

Studiengang Medieninformatik

Projektdokumentation

Projektbetreuer: Prof. Dr. Jens-Uwe Hahn

Wintersemester 2017/18

VR Classroom

Praktikum Virtual Reality

Verfasst von:

Lisa-Sophie Kramer

Matrikelnummer: 29971

E-Mail: lk089@hdm-stuttgart.de

Stuttgart, Februar 2018

Inhaltsverzeichnis

1. Einführung
2. Verlauf des Projektes und sich ergebende Aufgaben
3. Herangehensweise
4. Schwierigkeiten und besondere Herausforderungen
5. Projektstand am Ende des Semesters
6. Ausblick

1. Einführung

Das Projekt „VR Classroom“ fand im Rahmen der Vorlesung „Praktikum Virtual Reality“ im Wintersemester 2017/18 statt. Es handelt sich dabei um ein Forschungsprojekt, das in der Kooperation mit der Universität Tübingen entstand. Das Ziel war es ein virtuelles Klassenzimmer nach realem Vorbild zu erstellen.

Die Idee zu diesem Projekt stammte von der Universität Tübingen, deren Mitarbeiter diese beim Kick-Off Treffen zum „Praktikum Virtual Reality“ präsentierte.

Die Nutzung von Virtual Reality bietet einige Möglichkeiten, die für die Forschung sehr interessant sind, wie beispielsweise die Möglichkeit Szenen einfach zu reproduzieren und enthaltene Elemente einfach auszutauschen. Dies ist in der Realität nicht einfach möglich, da es dort viele Faktoren, die nicht beeinflussbar sind, gibt.

Da an der Universität Tübingen nicht das grundlegende Wissen sowie die für so ein Projekt benötigten Leute vorhanden waren, ging der Auftrag für eine Kooperation an die Hochschule der Medien.

Das Ziel des Gesamtprojekts ist die Erstellung eines grundlegenden Setups das für zukünftige Tests genutzt werden kann. Dieses soll auch erweiterbar und ausbaubar sein. Die grundlegende Erkenntnis, die zunächst aus dem Projekt gewonnen werden muss, ist ob Ergebnisse, die in der Realität erzielt werden in VR reproduzierbar sind. Im Umkehrschluss soll nachgewiesen werden das Ergebnisse, die aus VR gewonnen werden auch auf die Realität übertragbar sind.

Allgemein gilt es zudem herauszufinden, ob sich Virtual Reality für Studien dieser Art eignet.

In diesem Semester bestand das Ziel des Projektes in der Realisierung eines einfachen Testszenarios.

2. Verlauf des Projektes und sich ergebende Aufgaben

Zu Beginn des Projektes war die konkrete Aufgabenstellung noch unklar. Es war zunächst nur bekannt, dass ein virtuelles Klassenzimmer erstellt werden sollte, indem verschiedene Testszenarien durchlaufen werden können. Wie das Klassenzimmer aussehen bzw. ein Testszenario ablaufen sollte war anfangs noch unbekannt. Da zu Beginn des Semesters das Projekt noch nicht sehr ausgearbeitet war und auch noch einige organisatorische Dinge für die Zusammenarbeit geklärt werden mussten.

Weil jedoch keine Vorerfahrung auf Seite der Studenten vorhanden war, bestand die erste Aufgabe in der Einarbeitung in die Unreal Engine bis weitere Details bekannt waren.

Während die Einarbeitung in die Unreal Engine erfolgte, wurde parallel hierzu mit der Universität Tübingen ein Termin für einen Besuch vor Ort abgestimmt. Für diesen Besuch dort standen die Besichtigung des Klassenzimmers, das als Vorbild dienen sollte, und die genauere Definition des zu erarbeitenden Testszenarios auf dem Plan.

Während des Besuchs an der Universität Tübingen im November 2017 wurde ein einfaches Testszenario, das während des Semesters umgesetzt werden sollte definiert. Wichtig für das Testszenario war, dass dieses sowohl in der Realität als auch in Virtual Reality funktionieren sollte. Ebenso musste es auch in beiden Fällen umsetzbar sein, da die Ergebnisse beider Tests vergleichbar sein sollten.

Das Testszenario sollte wie folgt ablaufen. Zunächst läuft auf einem Bildschirm vorne im Klassenzimmer ein unterhaltsames Video, damit sich der Proband an die unbekannt virtuelle Umgebung gewöhnen kann. Auf dieses Video folgt ein Lehrvideo über ein uninteressantes Thema. Das unbekannt Thema soll den Probanden motivieren sich durch das unbekannt Umfeld ablenken zu lassen und dieses genauer zu betrachten. Nach dem Ende des Videos wird auf einem Laptop, der auf einem Tisch vor der Testperson steht, ein Multiple Choice Test angezeigt, den diese beantworten muss. Es wurde auf einen normalen Fragebogen verzichtet, da die Grundvoraussetzung war, dass der Nutzer den Test nur mit der Maus bedienen kann.

Durch das vorgegebene Testszenario konnten nun die einzelnen Elemente bestimmt werden, die umgesetzt werden mussten. Dabei handelte es sich um die folgenden Punkte:

- Einbindung eines 360°-Videos, das die Umgebung des realen Klassenzimmers mit den dort enthaltenen Personen simuliert
- Einbindung eines normalen Videos, das auf einem Bildschirm läuft
- Bau eines virtuellen Laptops, auf dem mit einer Maus interagiert werden kann um bspw. Fragen zu beantworten
- Einfacher Austausch von Elementen auf Knopfdruck

Gegen Ende des Projektes wurde noch als zusätzliche optionale Aufgabe der Bau eines Menüs hinzugefügt. Dies sollte nicht für den VR Nutzer sichtbar sein und von dem angeschlossenen Gerät aus bedient werden, um beispielsweise die Tastenbelegung festlegen zu können, mit der ein bestimmtes Element ausgetauscht wird.

Kurz vor der Media Night stand der Bau der finalen Szene an, in der die Modelle sowie alle erarbeiteten Komponenten zusammengefügt werden mussten.

3. Herangehensweise

Allgemeine Einarbeitung

Für die Einarbeitung in die Unreal Engine war der erste Anlaufpunkt die offizielle Dokumentation, in der einige Einsteiger-Tutorials zu finden waren. Da anfangs nicht bekannt war, wie die fertige Szene aussehen sollte, wurden zunächst nur allgemeine Tutorials durchgearbeitet, um die Engine kennenzulernen. Einen besonderen Fokus wurde hier jedoch bereits auf die sogenannten Blueprints für das visuelle Programmieren gelegt. Da diese später für die Hauptprogrammierung der Anwendung genutzt werden sollten.

Bereits recht früh war klar, dass ein Video eingebunden werden sollte. So war dies bereits die erste konkrete Aufgabe. Ebenso wurden in dieser Anfangsphase, die ersten Erfahrungen mit dem Einsatz von VR in Kombination mit der Unreal Engine gesammelt.

Allgemeines Vorgehen bei einem unbekanntem Problem

Zu Beginn eines unbekanntem Problems wurde zunächst überlegt, ob dieses so komplex war, dass dieses sich nochmals in Teilprobleme untergliedern ließ. Danach war der erste Anlaufpunkt zur Lösung des Problems ein Blick in die Dokumentation der Unreal Engine, da diese in vielen Bereichen bereits sehr ausführlich war und auch Anleitungen enthielt.

Meist nicht genügte dies nicht. Deshalb war die Suche im Internet der nächste Schritt, da es bereits eine ausgeprägte Community zur Engine gab, in der oft bereits jemand ein ähnliches Problem und Lösung gefragt hatte. Die ähnlichen Probleme konnten oftmals einen ersten Ansatz liefern, um weiterzusuchen und neue Lösungsideen zu finden. Oft war es jedoch sehr aufwändig eine passende Lösung bzw. –ansatz zu finden und es musste viel ausgetestet und ausprobiert werden.

Einbindung eines normalen Videos

Die Einbindung eines Videos stellte keine besondere Herausforderung dar, da für dies bereits ein Eintrag in der Dokumentation vorhanden war.

Einbindung eines 360° Videos

Zunächst erfolgte eine grundlegende Recherche im Internet zu diesem Thema, das auf der Überlegung basierte, dass die Einbindung eines 360°-Videos ähnlich funktioniert, wie die eines normalen Videos nur eben nicht auf einer flachen Ebene. Diese Überlegung erwies sich als richtig, denn in den gefundenen Tutorials wurde eine Hohlkugel genutzt, auf deren Innenseite das Video abgebildet wurde. Dies lieferte zwar die Lösung, um das Video einzubinden, jedoch hatte dieses noch keinen räumlichen Effekt, sondern die Darstellung war immer noch recht eben.

Um den Eindruck einer räumlichen Umgebung zu bekommen, sollte in der Theorie jeweils eine Hälfte eines stereoskopischen 360°-Videos auf jeweils eine Hohlkugel abgebildet werden. Je eine der Kugeln sollte dann nur für ein Auge an derselben Stelle gerendert werden. Herauszufinden wie dies in der Praxis umsetzbar war, wurde an den anderen Studenten übertragen.

Erstellung einer Maus mit der auf einem virtuellen Laptop interagiert werden kann

Zunächst wurde das Grundproblem in seine verschiedenen Bestandteile aufgeteilt. Dies waren die Folgenden: ein Element mit dem interagiert werden kann und auf dem Dinge darstellbar sind sowie eine Maus, die man bewegen können muss und mit der man mit dem vorherigen Element interagieren kann.

Das interaktive Element ließ sich durch die Nutzung eines sogenannten 3D-Widgets lösen. Dabei handelt es sich um ein Widget, das nicht direkt auf dem Bildschirm des Nutzers angezeigt wird, sondern als Objekt im Raum steht.

Der Bau der Maus stellte sich als aufwändiger als zunächst gedacht heraus. Da sich bei der Nutzung des Standardplayers, den die Unreal Engine zur Verfügung stellt, bereits anfangs ein grundlegendes Problem ergab. Wenn die Maus benutzt wurde, konnte man mit ihr das Sichtfeld in der VR zusätzlich zu der eigentlichen Bewegung durch den Kopf drehen.

Da dies nicht gewollt war, musste ein Pawn, eine unbewegliche Figur, als Spieler genutzt werden. Im Nachhinein stellte diese Lösung sich sogar als der empfohlene Ansatz in der Dokumentation heraus. Diese empfahl dies zu nutzen, wenn ein sitzender Nutzer die VR-Anwendung bedienen soll.

Der nächste Punkt, der angegangen werden musste, war die Darstellung der Maus. Zunächst erfolgte eine Grundüberlegung über die Eigenschaften des Mauszeigers. Er sollte nur auf einer beschränkten Fläche sichtbar sein, das Mausgerät sollte den sichtbaren Mauszeiger bewegen und der Bildschirm musste mit der Maus scrollbar sein. Der Mauszeiger sollte zudem als Hilfsmittel für Interaktionen auf dem Bildschirm verwendet werden können.

Der erste Ansatz war es den Standardmauszeiger des Computers zu nutzen und diesen zusätzlich in VR anzuzeigen. Jedoch ließ sich damit nicht die Sichtbarkeit der Maus auf eine definierte Fläche beschränken, da dieser Mauszeiger auf dem gesamten Blickfeld sichtbar war.

Der erste Lösungsansatz für dieses Problem war zunächst die Maus nicht anzuzeigen. Wenn der User auf das 3D-Widget blickt, mit dem er interagieren möchte, wird dies erkannt und der Mauszeiger wird angezeigt. Verlässt der Blick des Users das Widget wieder, soll der Mauszeiger wieder unsichtbar werden.

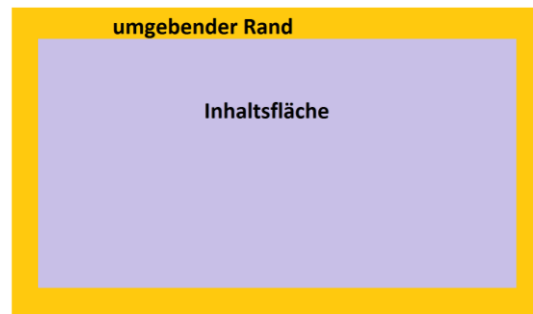


Abb. 1: Darstellung des Widgetaufbaus zum Auslösen der Maussichtbarkeit

Hierfür wird die Trace Line, die Blicklinie des Spielers, genutzt. Wenn diese den umgebenden Rand nach innen überschreitet, wurde die Maus sichtbar. Wenn die Trace Line den umgebenden Rand nach außen hin verlässt, wird die Maus unsichtbar.

Mit dieser Lösung konnte zwar die Zone eingeschränkt werden, in der der Mauszeiger sichtbar wurde, jedoch konnte der Mauszeiger immer noch über das gesamte Blickfeld bewegt werden. Dies schloss diesen Lösungsansatz aus. Zudem lieferte er die Erkenntnis, dass die Nutzung des normalen Mauszeigers sehr komplex ist und es schwer ist diesen einzuschränken.

Durch den zuvor gescheiterten Lösungsansatz entstand die Idee, ob es nicht möglich wäre einen eigenen Mauszeiger zu bauen, der die erforderlichen Kriterien erfüllt. Nach weiteren Nachforschungen wurde ein ähnlicher Ansatz ausfindig gemacht, in dem der Entwickler einen eigenen Mauszeiger gebaut hatte. In seiner Lösung nutzte er ein Bild auf einem Widget, das durch die Bewegung der Maus auf dem Widget bewegt wurde. Dieser Ansatz löste damit das vorherige Problem. Der Mauszeiger war damit nur noch in einem bestimmten Bereich sichtbar.

Dieser Lösungsansatz ließ sich ohne große Schwierigkeiten umsetzen. Jedoch ergaben sich noch einige kleinere Probleme, unter anderem sollte der Mauszeiger auf dem Widget am Rand sichtbar bleiben, auch wenn die Position des normalen Zeigers sich außerhalb des Widgets befand.

Durch zwei Funktionen, die berechnen, ob die Maus in x/y-Richtung auf der neuen Position sichtbar ist, durch das Abfragen dieser Sichtbarkeit und das entsprechende Handeln bevor der Mauszeiger neu positioniert wurde, ließ sich das vorherige Problem lösen.

Zunächst wurden die Sichtbarkeitsgrenzen nur anhand des sichtbaren Widgets berechnet, ohne Rücksichtnahme auf die variable y-Grenze des Widgets, wenn dieses gescrollt wurde. Später wurde dies jedoch hinzugenommen.

Zuletzt stand die Interaktion zwischen dem Mauszeiger und einem Element des Widgets zur Umsetzung an. Hier ergab sich das Problem, dass der tatsächliche Mauszeiger, der nativ zur Erkennung eines Klicks auf ein Element genutzt werden würde und dort das entsprechende Event auslösen würde, sich beim Klick auf das Element nicht über diesem befand. Jedoch

musste die Position des sichtbaren Mauszeigerbildes als Grundlage für die Auslösung des Klicks dienen.

Um dies zu erreichen, wurde nun bei einem Mausklick des Spielers, dies in einer Variable gespeichert und im Widget ein entsprechendes Event ausgelöst, das überprüfte über welchem Element das Mauszeigerbild sich in diesem Moment befand und löste dort die entsprechende Funktion aus.

Austausch einzelner Elemente

Zur Vereinfachung wurde beschlossen, dass es sich beim Austauschen eines Elementes anfangs auch um eine Texturänderung handeln kann und später dies auch ganze Elemente sein können.

Im Laufe des Projekts wurden folgendes Vorgehen genutzt, um die Textur auszutauschen. Zunächst wurde eine dynamische Materialinstanz erstellt, deren Textur auf Knopfdruck dann verändert wurde. Die entsprechende Taste wurde im Blueprint des Objektes festgelegt. Wichtig hierbei war, dass auf dem Objekt, dessen Textur verändert werden sollte, Userinput möglich gemacht wurde. Festgelegt, welche Textur der dynamischen Materialinstanz zugeordnet wurde, wurde durch die Nutzung einer Enumeration vereinfacht.

Dieses Vorgehen stellte sich später in dem Punkt als schlecht heraus, dass der User Input direkt im Blueprint des Objektes angelegt war. Dies wurde dann durch das Anlegen eines Custom Events und dessen Zuweisung im Blueprint des Spielers korrigiert.

Bau eines Menüs

Die Grundschwierigkeit für das Menü bestand darin, dass es nicht für den Nutzer sichtbar sein sollte, der die VR Brille trägt. Es sollte nur auf dem Bildschirm des angeschlossenen Computers sichtbar sein.

Den Beginn machten Nachforschungen, um herauszufinden mit welchen Möglichkeiten dies umsetzbar war und welche Ansätze andere Leute, die etwas ähnliches bauen wollten bereits hatten.

Dabei kamen unter anderem folgende Lösungsansätze heraus:

- Es wird ein zweites Fenster geöffnet, in dem das Menü angezeigt wird. Dieses soll nur auf dem Computer angezeigt werden und damit unabhängig vom eigentlichen Spielfenster sein. Dies stellte sich jedoch ohne die Verwendung von C++ als kaum umsetzbar heraus. Da in C++ keinerlei Vorkenntnisse vorhanden waren, wurde dieser Lösungsansatz nicht gewählt.
- Der zweite Ansatz war die Nutzung eines Widgets, das nur auf dem Computerbildschirm angezeigt wird. Während dieses angezeigt wird, sollte das Bild der VR Brille entweder abgeschaltet oder auf schwarz gestellt werden. Während des

Experimentierens mit diesem Ansatz stellte sich bald heraus, dass die Trennung zwischen dem Bild der VR Brille sowie dem Bild des Computerbildschirms nicht so ohne weiteres möglich war.

- Auf der Suche nach einer Lösung für das Problem, das sich im zweiten Ansatz herausstellte, wurde das noch relativ neue Feature des Spectator Modes entdeckt. Hierbei wird eine zweite Kamera genutzt, die durch die Aktivierung des Spectator Modes, dieses Bild nur auf dem Computerbildschirm, dem Spectator Screen, anzeigt.

Die Entscheidung fiel auf die Nutzung des Spectator Modes. Dieses Feature wurde zunächst nach der Anleitung einiger Tutorials ausgetestet. Prinzipiell sollte dieser Mode genutzt werden können, um zum Einen den Spieler in VR beobachten zu können sowie in anderen Szenarien auch die virtuelle Umgebung zu beeinflussen. Jedoch stellte sich schnell heraus, dass der Punkt „Beeinflussung“ noch nicht wirklich ausgereift war.

Eines der Grundprobleme war unter anderem, dass nur ein Weg gefunden wurde, eine 2D-Textur auf einer sogenannten Spectator Camera, die für den Spectator Mode genutzt wurde anzuzeigen. Mit dieser 2D-Textur war damit keine Interaktion möglich. Auf der Suche nach Lösungen wurde ein Ansatz gefunden ein Widget als 2D-Textur auf die Kamera mithilfe eines Plugins zu rendern. Dies konnte jedoch nicht ausgetestet und beendet werden.

4. Schwierigkeiten und besondere Herausforderungen

Allgemeine Schwierigkeiten

Unter die allgemeinen Schwierigkeiten, die während der Durchführung des Projektes angefallen sind, fällt, dass die Kommunikation mit der Universität Tübingen sehr lang dauerte. Besonders zum Anfang des Projekts dauerte es einige Zeit bis feststand was gewollt war und was im Laufe des Semesters erarbeitet werden soll. Dies führte anfangs nicht nur zu Verzögerungen, sondern auch zu einer sehr langen Einarbeitungsphase.

Zudem gab es einige Probleme in der Zusammenarbeit mit dem anderen Studenten, der sich für das Projekt gemeldet hatte. Dieser kam nicht zu wöchentlichen Treffen und war auch im Allgemeinen sehr unzuverlässig in seiner Arbeit, da er selten Ergebnisse und Fortschritte vorzuweisen hatte. Besonders am Ende des Projekts, als die Endszene gebaut wurde, konnte aus diesem Grund ein Element nicht eingebunden werden.

Zu den organisatorischen Schwierigkeiten kamen vor allem zu anfangs und auch im späteren Verlauf noch technische in der Nutzung mit der Oculus DK2 hinzu.

Zunächst dauerte es einige Zeit bis die Einrichtung der Brille abgeschlossen war. Dies lag einerseits an der Treibersoftware, die von Oculus bereitgestellt wird, andererseits an verschiedenen Konfigurationsproblemen und Inkompatibilität zwischen verschiedenen Windows-Treibern und der Brille.

Gegen Ende des Projekts wurden eben jene Treiber zum Problem, sodass an dem ausgeliehenen Laptop der HdM, der HDMI-Anschluss der Brille nicht mehr erkannt wurde. Dies schränkte zu diesem Zeitpunkt die Weiterarbeit am Projekt massiv ein. Auch nach Rücksprache mit dem Oculus Support, konnte die Brille nicht wieder zum Laufen an dem Laptop gebracht werden. An anderen Rechnern trat dieses Problem jedoch nicht auf.

Einbindung des 360°-Videos

Es gab einige Schwierigkeiten bei der Einbindung des 360°-Grad Videos. Das Einbinden selbst funktionierte zwar. Jedoch ergaben sich Probleme eine Lösung zu finden, um zwei Hohlkugeln auf jeweils einem Auge zu rendern. Von Haus aus war dies nicht einfach möglich, da man hierfür mit C++ Code hätte arbeiten müssen und keine Nutzung von Blueprints möglich war.

Da ein anderes Projektmitglied an diesem Thema gearbeitet hat, kann zu den Schwierigkeiten an dieser Stelle nicht noch mehr gesagt werden.

Bau der virtuellen Maus

Eine generelle Schwierigkeit trat bei der Nutzung der realen Maus in der Unreal Engine für die geplanten Zwecke auf. Genauerer hierzu findet sich im Abschnitt Herangehensweise mit den jeweiligen Lösungsansätzen für die aufgetretenen Probleme.

Hierzu zählte, dass wenn man mit dem normalen Mauszeiger an die Ränder des Spielfensters gelangte das Bild des Mauszeigers im Widget viel herumsprang. Eine Abhilfe hierfür war die Nutzung eines Fullscreen-Windows.

Ein Grundproblem war außerdem an dieser Stelle, dass eine lange Suche nötig war, um den genutzten Lösungsansatz zu finden, da eigenen Ideen unter anderem dadurch eingeschränkt waren, dass auch nicht das Wissen über verschiedene Möglichkeiten vorhanden war. Zudem mussten einige Berechnungen des Öfteren korrigiert werden, da es sehr viele Fälle gab, die bedacht werden mussten, wenn der Mauszeiger bewegt wurde. Beispielsweise zählten hierzu, ob der Inhalt bereits gescrollt war und ob die Position des Mauszeigers sich noch im sichtbaren Bereich befand.

Schwierigkeiten gab es zudem bei der Interaktion zwischen dem Widget und dem Bild des Mauszeigers. Zunächst funktionierte die Berechnungsfunktion nicht und die Gründe waren hierfür nicht erklärbar, bis durch Zufall herausgefunden wurde, dass die Elemente des Widgets im umgebenden Canvas gleichausgerichtet werden müssen. Ebenso wurde noch keine gute Lösung gefunden, die Klickfunktionen automatisch auszulösen. Momentan müssen diese noch manuell den einzelnen Elementen des Widgets zugewiesen werden.

Nutzung des Spectator Modes der Unreal Engine

Das noch relative neue Feature des Spectator Modes brachte ebenfalls einige Schwierigkeiten mit sich. Zwar war die prinzipielle Nutzung sehr einfach und ein erstes Setup ließ sich nach der Dokumentation gut erstellen.

Jedoch sind seine Funktionen noch sehr eingeschränkt und es war zum Zeitpunkt des Projektes nur möglich 2D-Texturen in einer Spectator Camera zu rendern. Durch diese Einschränkung war es nicht einfach so möglich ein Widget dort zu rendern, mit dem auch interagiert werden konnte. Hierfür wurde eine mögliche Lösung in Form eines Plugins gefunden, die jedoch aufgrund technischer Schwierigkeiten am Ende und zeitlicher Beschränkung nicht mehr ausgetestet werden konnte.

5. Projektstand am Ende des Semesters

Am Ende des Semesters wurden die Elemente, die für das definierte Testszenario erarbeitet wurden in eine Szene hineingearbeitet, die ein virtuelles Klassenzimmer darstellte. Dieses war angelehnt an das reale Klassenzimmer, das in der Universität Tübingen besichtigt wurde.



Abb. 2: Virtuelles Klassenzimmer am Ende des Semesters

Die Szene stellt ein einfaches Klassenzimmer mit Stühlen, Tischen, auf denen Laptops stehen und einem Bildschirm an der Wand dar, die von Benjamin Wohlbrecht modelliert wurden. Auf dem Bildschirm an der Wand läuft ein Video ab. Auf dem Laptop, der auf dem Tisch vor dem sitzenden Nutzer des VR Geräts steht, kann eine virtuelle Maus bewegt werden, die durch eine reale bedient wird. Mit der Maus kann der angezeigte Text gescrollt werden, sowie auf einen Button geklickt. Durch die Nutzung einer Taste kann die Wandfarbe von weiß auf rot geändert werden.

Nicht enthalten sind in dieser Szene erstens ein 360°-Video, da zum einen nicht wirklich das passende Videomaterial, eine Aufnahme aus dem Testlauf in Tübingen, vorliegend war. Zum anderen war der zuständige Student, der sich über das Semester hinweg mit diesem Thema hätte beschäftigen sollen, nicht anwesend als die finale Szene zusammengesetzt wurde. Zweitens konnte das Menü nicht fertiggestellt werden, da es noch einige Schwierigkeiten mit dem verwendeten Feature der Unreal Engine gab und die Zeit nicht mehr ausreichte.

Während der Media Night trat ein Fehler auf, dass das Video nicht automatisch abgespielt wurde, wenn das Projekt nicht im Vorschaumodus gestartet wurde. Dieser wurde noch nicht

behooben und somit musste auf der Media Night mit dem Vorschaumodus gearbeitet werden.

6. Ausblick

Das Projekt selbst soll im kommenden Semester fortgeführt werden. Zudem wurde ein neuer Mitarbeiter speziell für das Forschungsprojekt eingestellt, damit dieses ausgebaut wird. Das erarbeitete Grundsetup soll erweitert und die erarbeiteten Elemente können optimiert und angepasst werden.

Sobald das erste Testszenario und die entsprechenden Materialien für das Video sowie den Fragebogen stehen können die ersten Tests stattfinden.

Eine mögliche Anpassung wäre die Verwendung eines 360°-Videos, das während eines Testlaufs in der Realität aufgenommen und das bisher noch nicht eingebunden wurde, sowie die Fertigstellung der optimalen Einbindung dieses. Ebenso könnten reale Charaktere modelliert werden, die sich anstelle des Videos im Raum befinden und Mitschüler simulieren. Ebenso könnten weitere Elemente hinzugefügt werden, die auf Knopfdruck geändert werden. Auch könnte ein Knopfdruck auch verschiedene andere Dinge auslösen, wie ein gezieltes Störgeräusch.

Es stehen noch viele Möglichkeiten der Erweiterung des Projektes offen und es wird noch etwas Arbeit benötigt werden, bis dieses in den ersten Studien zum Einsatz kommt. Der Grundansatz jedoch, die Virtuelle Realität für die Forschung zu verwenden, stellt einen interessanten Grundgedanken dar, der zwar noch sehr unerprobt ist, jedoch sehr viel Potential bietet, um erforscht zu werden und erste Projekte zu starten. Spannend wird es, sollte sich erweisen, dass das Grundziel – die Übertragung von Erkenntnissen aus der VR in die Realität – möglich ist, könnte sich daraus ein interessantes Forschungsgebiet ergeben. Ebenso würde dies einige Prozesse vereinfachen, die momentan noch eine sehr große Organisation benötigen.