

Motion Capture for E-Learning Entwicklung

Lali Nurtaev Tristan Schmele Najla Barkallah

Modul: Medieninformatik - Master
Kurs: Projekt II - Entwicklung

Projektleiter: Prof. Dr. Horst Stenzel
Betreuer: Jannis Möller

September 30, 2022

Contents

1	Einleitung	3
2	Vorbereitung	4
2.1	Domänenmodell	4
2.2	Seitenstruktur	5
2.3	Styleguide	6
2.4	Architekturskizze	9
3	Frontend	11
3.1	Umsetzung mittels HTML/CSS	11
3.2	Umsetzung mittels Vue	11
4	Datenbank	14
4.1	Dokumentbasierte Datenbanken	14
4.2	Spaltenorientierten Datenbanken	14
4.3	SQL Datenbank	14
4.4	Datenbankstruktur	15
5	Video Player	16
5.1	Ziele und Erwartungen	16
5.2	Auswahl des Frameworks	17
5.3	Umsetzung mit Unity	17
5.4	Evaluation und Potential	20
6	Motion Capture	21
7	Minimum Viable Product	23
8	Review	25
9	Fazit	26
	Abbildungsverzeichnis	27
	Listingsverzeichnis	28
	Literaturverzeichnis	29
A	Anhang	30
A.1	MVP	30

1 Einleitung

Im Verlauf dieses Projekts, führt das Team die Implementierung der, in einem vorangegangenen Projekt konzipierten, e-Learning Plattform mit Motion Capture durch. Als teil dieser Plattform werden ein Video Player zur Darstellung und Interaktion mit 3D Szenen, so wie die Webanwendung, in welcher dieser Video Player genutzt werden soll, umgesetzt.

Die finale Plattform wird zwei Hauptbereiche umfassen, einen für Kursleiter, in welcher diese auf ihre Motion Capture Aufnahmen zugreifen können und mit diesen Lehrvideos erstellen können und einen für die Kurse in welchen die Lehrvideos geteilt werden und von Teilnehmern betrachtet werden können. Das Ziel für dieses Projekt ist die Erstellung eines Minimum Viable Products (MVP), welches sich ursprünglich auf den Videoeditor für Kursleiter konzentrieren sollte, wurde jedoch im Verlauf des Projekts geändert und umfasst nun die Kurse und das anschauen von Videos als Kursteilnehmer. Der MVP soll eigene Motion Capture Aufnahmen als Beispiel verwenden.

Um dieses Ziel zu erreichen, wird zu Beginn nach weiterer Recherche der Domäne eine Grundstruktur erstellt mit welcher die Webanwendung umgesetzt werden soll. Anhand dieser Struktur werden der Video Player und die Webanwendung weitestgehend unabhängig realisiert und anschließend im MVP kombiniert.

Figur 1 stellt den Soll-Zustand der Domäne dar unter Verwendung des e-Learning Systems. Weitere Schritte des Projektes nutzen dieses Modell als Grundlage. Hervorzuheben in diesem Modell sind die zwei Spezialisierungen von Benutzer “Kursleiter” und “Teilnehmer”, welche beide mit einem oder mehreren Kursen zusammenhängen, aber stark unterschiedliche Rollen erfüllen. Der Kursleiter hat eine direkte Beziehung zu sowohl Motion Capture Clip, als auch zu Video. Dies zeigt die Aufgabe des Kursleiters, Motion Capture Aufnahmen zu organisieren und diese Aufnahmen zu einem Lehrvideo zu kombinieren. Die Kursteilnehmer sind hauptsächlich passive Akteure im System. Sie verlassen sich darauf, dass Kursleiter die Videos erstellen und zur Verfügung stellen und lernen mit diesen die Themen des Kurses. Die vorerst einzige aktive Interaktion mit dem System für Kursteilnehmer ist über Fragen oder Feedback in Form von Kommentaren zu einzelnen Videos.

Nicht in diesem Modell enthalten sind alle Geschäftsobjekte bezüglich dem erstellen von Motion Capture Aufnahmen. Dieser gesamte Prozess wurde bewusst aus dem Rahmen des Projekts ausgeschlossen, da die Methode der Aufnahme derzeit nicht das System betrifft und keinen Einfluss auf die weitere Entwicklung hat.

2.2 Seitenstruktur

Für eine weitere Entwicklung des Projekts wurde eine Seitenstruktur erstellt. Ein gründlich durchdachter Aufbau der Website ist essenziell. Einerseits spielt eine logische Seitenstruktur für die Nutzerfreundlichkeit der Seite eine große Rolle. Andererseits ist der Seitenaufbau auch für die Suchmaschinenoptimierung von Bedeutung.

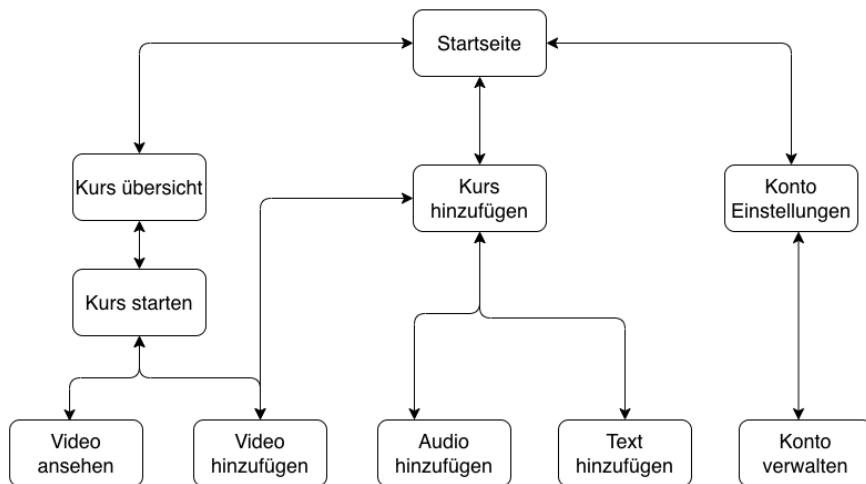


Figure 2: Modell für die Seitenstruktur der webseite

Wie in Figure 2 zu sehen ist, ist eine Navigation zwischen der verschiedenen Seiten möglich. Aus der Startseite steht ein direkter Zugriff auf Konto Einstellungen. Der Benutzer ist auch in der Lage eine Kursübersicht zu haben

sowie einen neuen Kurs hinzufügen. Die Beziehung zwischen der Seiten ist symmetrisch. Das heißt die Navigation ist deutlicher einfacher gestellt.

2.3 Styleguide

In der Konzeptionsphase im 1. Semester wurde bereits ein Prototyp in kurzer Zeit erstellt. Für die Entwicklung des User Interface (UI) wird zunächst ein Styleguide erstellt, um bei der Entwicklung des Frontends eine schnelle Implementation zu ermöglichen. Der Styleguide wurde iterativ erstellt, wobei die erste Version einen groben Überblick bieten sollte, welche Elemente das Frontend haben sollte und wie sie dargestellt werden. Die zweite Version enthält angepasste Elemente, da sich während des Projekts die Zielsetzung geändert hat. Im ersten Styleguide wurde eine Farbpalette mittels der Seite Coolors

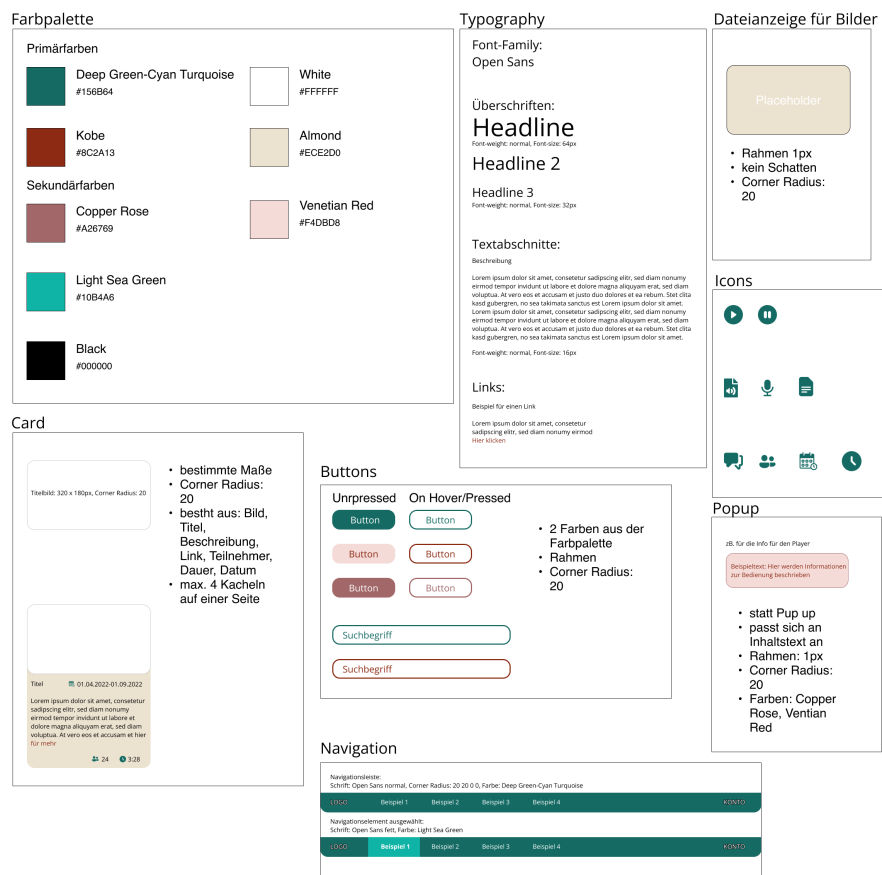


Figure 3: Styleguide Version 1

generiert, um mehrere Farben, die zueinander passen zu definieren (siehe Figure 3). Diese wurden in Primärfarben, die die wichtigsten UI Komponenten markieren sollen, und in Sekundärfarben, die weitere Elemente kennzeichnen und Kontraste vermitteln, eingeteilt. Weiterhin wurde die Typografie definiert.

Diese beinhaltet die Font-Family und je nach Überschrifts- oder Bodytext, die Font-Größe und das Font-Gewicht.

Ein wichtiges Element sind Cards, die einen Kurs mit den wichtigsten Inhalten darstellen. Die Entscheidung für die Nutzung von Cards wurde getroffen, da sie Eingrenzungen zusammen gehörender Inhalte wie Kursbild, -titel, -datum, -beschreibung, etc. präsentieren. Innerhalb der Cards werden Informationen mittels Icons vermittelt, da das Gehirn Grafiken schneller verarbeiten kann und so beispielsweise die Teilnehmeranzahl einer Gruppe sofort innerhalb der Card gefunden werden kann.

Ebenso wie die Card sind anhand der Farbpalette Elemente wie Navigation, Suchleiste und Button entstanden, die alle einen einheitlichen Corner Radius von 20px haben, um ein modernes Aussehen zu erhalten. Es wurden zudem Zustände beim Hover definiert für eine bessere Benutzerfreundlichkeit und um dem Nutzer zu symbolisieren, welche Elemente angeklickt werden können.

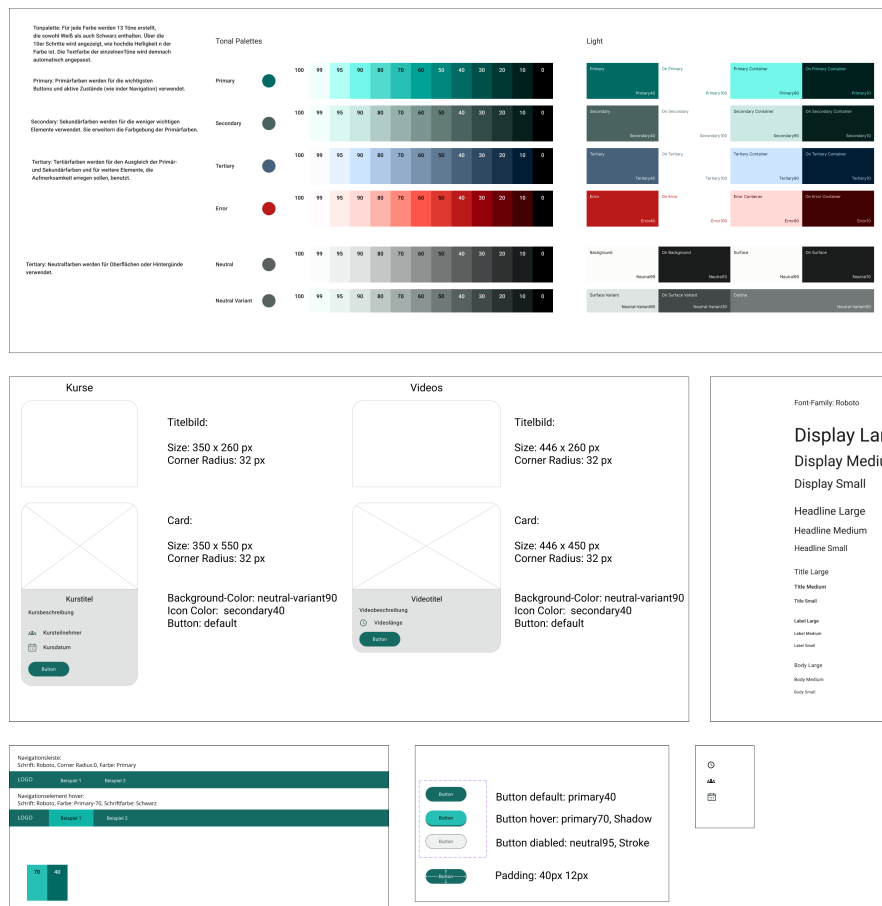


Figure 4: Styleguide Version 2

Der zweite Styleguide (siehe Figure 4) wurde mithilfe der Designsprache Material Design (MD) von Google erstellt Google (2022). Das MD ist inspiriert von der realen Welt, da dort Schattierungen und Farben reflektiert werden. Diese Iteration war notwendig, da in der ersten Version kein Wert auf Benutzerfre-

undlichkeit oder Barrierefreiheit gelegt wurde. Zudem erleichtert das MD die Erstellung eines Styleguides, da nicht nur Farbpaletten sondern auch die Typografie automatisch generiert werden und eine Ansammlung bestimmter Icons angeboten wird. Das MD berücksichtigt auch die Barrierefreiheit im Web, da der Farbkontrast in einer Farbe in der Farbpalette und Textfarbe immer angepasst ist. So hat die angepasste Navigation beim Hovern eines Elementes eine hellere Hintergrundfarbe, jedoch auch eine schwarze Schrift.

In der Iteration wurde das Element Card überarbeitet, da es zum einen zu klein war und zum anderen soll der Nutzer durch eine Erhebung der Card beim Hover sehen, dass es ein anklickbares Element ist. Die Card enthält dennoch weiterhin die wichtigsten Informationen, wie ein Titelbild, Titel, Beschreibung und Kursinformation mit Icons. Die Icons sind ebenfalls aus dem MD importiert. Zudem wird der Nutzer durch einen Button zu einem Kurs oder Video geleitet, da dieser aussagekräftig für den nächsten Screen ist.

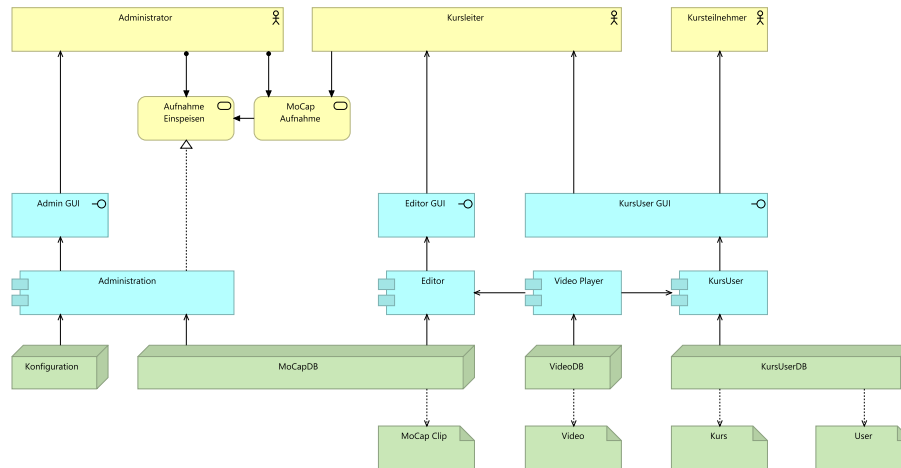


Figure 5: Architektur Skizze. Zeigt die wesentlichen Elemente des e-Learning Systems unterteilt (von oben nach unten) in Geschäfts-, Applikations- und Technologie-Ebene.

2.4 Architekturskizze

Um die Planung und Implementierung des MVP strukturiert durchführen zu können, wurde basierend auf dem Domänenmodell die in Figur 5 zu sehende Architektur skizziert. Die zwei Spezialisierungen von Benutzer wurden durch eine allgemeine Administrator-Rolle ergänzt und sind die Akteure in diesem Modell. Die einzelnen Bereiche des Systems sollen intern lose gekoppelt sein, um bei späterem Bedarf eine Migration zu einer Microservice-Architektur mit wenig Aufwand möglich zu machen.

In dieser Form sind Motion Capture Aufnahmen als Business Service modelliert und würden extern stattfinden, während das einspeisen der Aufnahmen in das System durch einen Administrator getätigt werden. Zusätzlich ist die Administrator Rolle für weitere Systemkonfiguration zuständig. Für den MVP wird dieser Aspekt vorerst vernachlässigt.

Die zwei Bereiche für Benutzer, der Editor für Kursleiter und der reguläre Kursbereich, welcher primär für Kursteilnehmer ist, sind durch zwei separate GUI Schnittstellen zugänglich und haben intern separate Applikationskomponenten. Der Video Player ist ebenfalls eine eigene Applikationskomponente und wird sowohl von Editor als auch von KursUser genutzt. Um eine zukünftige weitere Aufteilung in Microservices ohne große Änderung zu ermöglichen sind hier ebenfalls die drei Hauptkomponenten nur mit ihrer eigenen Datenbank verknüpft.

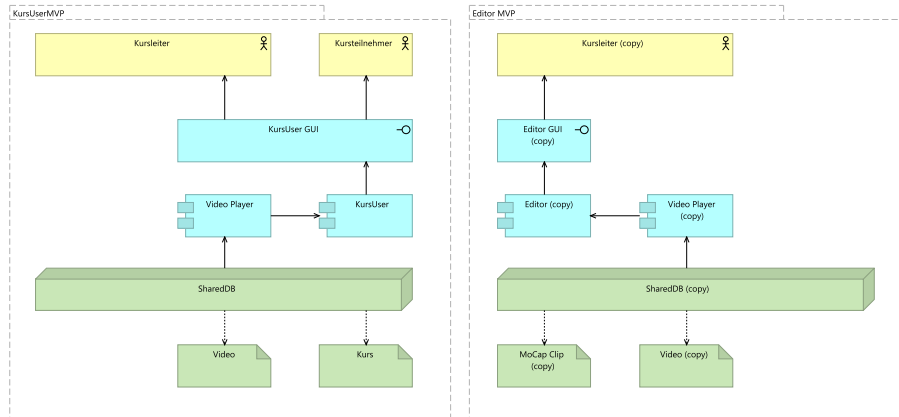


Figure 6: Modifikation der gesamten Architektur Skizze. Der KursUser MVP (links) enthält KursUser GUI und Applikationskomponente, der Editor MVP (rechts) enthält Editor GUI und Applikationskomponente. Beide MVP enthalten eine gesamte Datenbank und die Video Player Komponente.

Figur 6 zeigt die modifizierte Version der Architektur für die beiden Optionen des MVP. In beiden Versionen ist nur eine gesamte Datenbank für alle Komponenten des MVP vorgesehen, welche für einen vollen Release wieder aufgeteilt werden muss. Diese Version bildet die Grundlage für die weitere Arbeitsteilung und Planung des Projekts.

3 Frontend

Im folgenden Kapitel werden die einzelne Schritte zur Entwicklung der Frontend genannt.

3.1 Umsetzung mittels HTML/CSS

Zweck der Erstellung wurden HTML und CSS angewendet um die eingesetzte Ideen zu realisieren? Wie in KAP 2.3 beschrieben ist, wurden eigene Farben sowie Styles ausgewählt. Die Benutzeroberfläche sollte so viel wie möglich sauber und selbst erklärend sein.

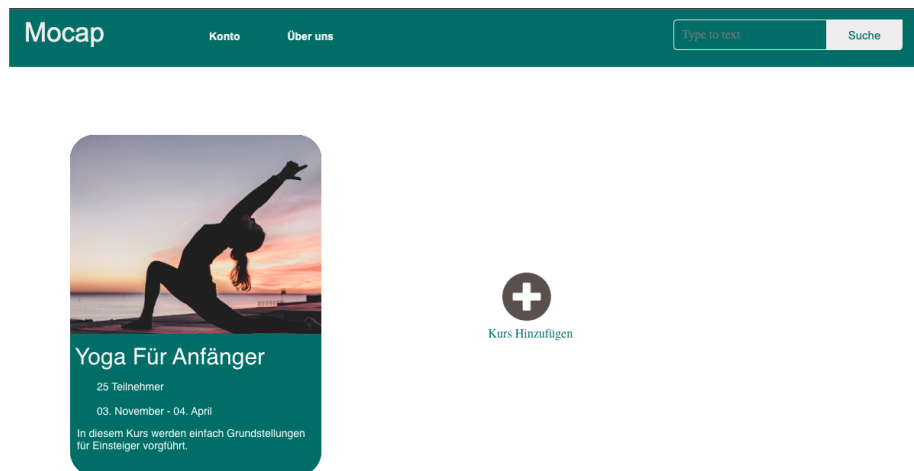


Figure 7: Die Home-Seite der erstellten Webseite

Die Figure 7 zeigt die Startseite auf der Verschiedene Optionen zur Verfügung stehen. Der Nutzer ist in der Lage auf Konto zuzugreifen sowie einen Kurs zu starten. Das hinzufügen eines Kurses erfolgt durch das Button, das in der Mitte steht. Verschiedene Informationen werden unter über uns publiziert.

Die Figure 8 zeigt die Seite der Videoplayer. Darauf kann der Nutzer Clips abspielen, um die verschiedene Tanzschritten kennenzulernen. Unter jeder Clip wird ein Text hinzugefügt, der die gezeigte Bewegungen auch schriftlich erklärt.

3.2 Umsetzung mittels Vue

Als Alternativen für die Erstellung einer Webseite wurden Webseiten-Baukästen wie Strato und Squarespace betrachtet. Der Vorteil dieser Optionen ist, dass die Erstellung einfach und schnell ist, da vordefinierte Templates angewandt werden. Jedoch haben wir uns dagegen entschieden, weil sie zum einen kostenpflichtig und zum anderen nur bis zu einem bestimmten Grad individuell anpassbar sind. So kann ein Videoeditor oder eine Timeline nicht eingefügt werden. Aus diesem Grund wurde sich dafür entschieden eine Webseite selber zu erstellen. Ein Vorteil hiervon ist, dass es ein geringer Aufwand ist, da generell

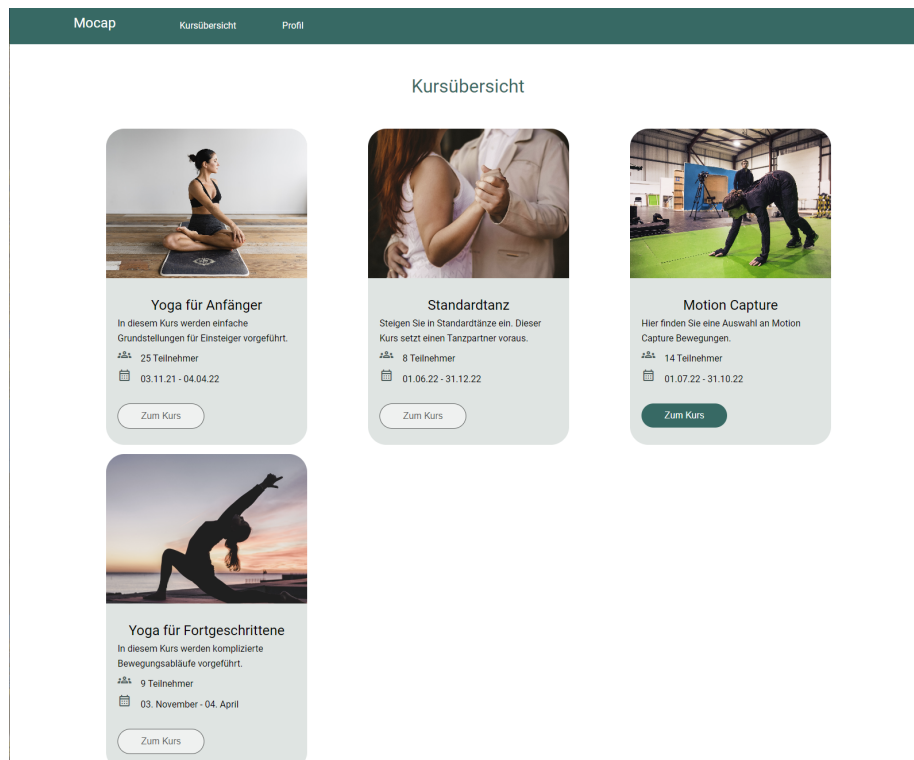


Figure 8: Die überarbeitete Home-Seite der erstellten Webseite

eine kleine Anzahl an Komponenten wie Timeline, Einbettung des Videoplayers und Dateiverzeichnis erstellt werden müssen.

Da für die Implementierung des Unity Videoplayer Fensters mittels WebGL ein Webserver benötigt wird, wird ein lokaler Webserver für das Projekt aufgesetzt. Für die Implementierung wird das JavaScript Framework Vue.js verwendet, da eine Web-Applikation, die auf Komponenten basiert, eine schnellere und flexiblere Lösung darstellt.

Für die Entwicklung standen auch React und Angular zur Auswahl, da sie sich genauso wie Vue eignen, jedoch wurde sich aus persönlichem Interesse für das Erlernen des Vue-Frameworks entschieden. Die in Kapitel 2.2 definierte Seitenstruktur lässt sich in Vue in einzelne Views umwandeln, die mithilfe eines RouterLinks bei einem Seitenwechsel aufgerufen werden. Die Designs des Styleguides aus der 1. und 2. Version sind übernommen worden, konnten jedoch nicht vollständig angewandt werden, da beispielsweise Informationen über Spacings zwischen Elementen im Style Guide nicht verfügbar waren.

Für die Einbettung des Unity Videoplayers wird die Bibliothek unity-webgl verwendet. Dafür werden die aus dem WebGL Export entstandenen Build-Dateien loader, data, framework und wasm des Unity-Projektes benötigt und im Code verlinkt. In dem HTML-Element Canvas wird das Unity-Projekt angezeigt. Die Steuerung, die in Kapitel 5 erläutert wird, ist somit in der Web-Applikation interaktiv bedienbar. Der Code für den Videoplayer ist angelehnt an folgenden Demo-Code. Mit den URLs aus dem Code wird die Datenextraktion aus einer

Datenbank in Kapitel 4 getestet.

4 Datenbank

Das Auswählen einer Datenbank ist jetzt eine Verpflichtung für die weitere Entwicklung des Projekts. Es ist wichtig, von Anfang an eine fundierte Entscheidung in Bezug auf Datenbankentechnologien zu treffen. Zu Beginn wurde es einige Informationen zu unterschiedlichen Datenbanktypen gesammelt, um eine fundierte Entscheidung zu treffen. Die Recherche ist auf Bases der Dokumentorientierten bzw. Spaltenorientierten Datenbanken unterschieden. Die unterschiedliche Datenbanken wurden analysiert um die Vor- und Nachteile der Anwendung zur verstehen und damit gesamte Arbeit effizienter und strukturierter zu haben.

4.1 Dokumentbasierte Datenbanken

Eine große Auswahl der Datenbanken steht zur Verfügung. Daher wurden einige Konzepte näher erklärt und analysiert.

CouchDB

CouchDB ist eine dokumentorientierte sowie schemafreie NoSQL-Datenbank. Im Gegensatz zu den verschiedenen Datenbanksystemen werden die Daten schemafrei gespeichert. Das bedeutet, dass keine Tabellen und Spalten sowie Fremdschlüssen und Beziehungen gibt, sondern werden die Daten in einer Reihe von Dokumenten gespeichert und es wird eine JavaScript-basiertes Ansichtsmodell zum Aggregieren und Berichten über die Daten angeboten. Lennon (2009)

MongoDB

MongoDB hat das Konzept von Tabellen; Schemas, SQL oder Zeilen. Es hat dabei auch keine ACID-Compliance oder Joins. MongoDB-Dokumente sind im Vergleich zu JSON-Objekten ähnlich. Die Werte von Feldern können andere Dokumente, Arrays und Arrays von Dokumenten enthalten. Subramanian (2019)

4.2 Spaltenorientierten Datenbanken

Unter der spaltenorientierten Datenbanken versteht man die Art, wie die Daten auf dem Speichermedium gespeichert sind. Mit Hilfe der sogenannten Datenbanksystem erfolgt die Speicherung der Daten spaltenweise und nicht wie bei relationalen Datenbanken zeilenweise. Eine Beschleunigung der Datenbereitstellung erfolgt dadurch, in dem die Festplattenzugriffe reduziert werden. Einen weiteren Vorteil von spaltenorientierten Datenbanken ist es die Möglichkeiten der Kompression. Jede Daten in einer Spalten sind immer vom gleichen Typ. Dies lässt das Komprimieren effizienter.

4.3 SQL Datenbank

Ist ein Datenbankmanagement System. Es bietet die Grundlage für viele dynamische Webauftritte. Es werden Datenbanken zu Verfügung gestellt, in denen Inhalte mittels Tabellen abgespeichert werden. Um auf die Daten zu zugreifen, kann zum Beispiel die Scriptsprache PHP verwendet werden.

4.4 Datenbankstruktur

Die Struktur der Datenbank wurde mithilfe des Domänenmodells erstellt. Die Entscheidung zur Realisierung fiel auf MySQL Workbench, weil die Teammitglieder bereits Kenntnisse mit dem Modellierungswerkzeug haben und die Erstellung schnell erfolgen kann. Tabellen können sowohl durch Zusammenklicken oder SQL-Befehle erstellt und Daten ebenso gespeichert werden. Ein weiterer Vorteil von MySQL Workbench ist das automatische Erstellen vom erweiterten Entity-Relationship Modell (siehe Figure 9). Da die Plattform jeweils eine Ansicht für Lehrkräfte und Schüler besitzt, werden für diese zwei Nutzergruppen jeweils Entitäten erstellt. Es werden Primärschlüssel und die wichtigsten Informationen wie Name, E-Mail-Adresse und Passwort als Attribute für ein Benutzerkonto gespeichert. Zudem wird eine Entität für Kurse definiert, indem neben den Kursinformationen auch ein Fremdschlüssel des Kursleiters gespeichert wird, da für jeder Kurs durch eine Lehrkraft geleitet wird. So wird die Beziehung zwischen den Entitäten festgelegt, dass ein Kursleiter ein oder mehrere Kurse haben kann. Eine weitere Entität ist Video hat Attribute die u.a. die vier Build-Dateien eines Unity-Projekts enthalten. Über diese Dateien wird der Inhalt des interaktiven Videoplayers festgelegt (siehe Kapitel 3). Hierbei gilt, dass ein Kurs ein oder mehrere Videos enthalten kann. Um die Beziehung zwischen der Entität Schüler und Kurs abzubilden, wird eine zusätzliche Entität "Kurs_hat_Student" erstellt, der einen zusammengesetzten Primär- und Fremdschlüssel vom Kurs und Schüler besitzt. Anhand dessen ist definiert, dass ein Kurs oder mehrere Kurse einen oder mehrere Schüler haben kann und umgekehrt.

Als Nächstes wurden Testdaten in die in die Tabellen geladen und eine Verbindung zur Datenbank in einem Node Express Projekt aufgebaut. Für jeden Kurs wurden die Kursdaten automatisch extrahiert. Ein Problem trat beim Laden des Kursbildes auf. Dieses wurde zunächst als Dateityp `LONGBLOB` gespeichert, da es sich zum Speichern von Dateien unterschiedlicher Art eignet. Das Problem war, dass die Binärdaten des Bildes in Node nicht als Bild umgewandelt werden konnten. Dies wurde gelöst indem der Dateityp in `VARCHAR(2048)` geändert wurde, um eine URL zu dem Bild zu hinterlegen. Mit dieser Option konnten die Kursbilder aus dem GitHub Repository automatisch geladen werden. Das selbe Problem trat bei den Attributen für die Build-Dateien des Unity-Projekts der Entität Video auf. Die Dateien wurden sowohl in GitHub, Sciebo als auch GoogleDrive abgelegt, jedoch konnten sie nicht abgerufen werden. Mit den URL-Links aus dem Code aus Kapitel 3 wurde das Unity-Projekt angezeigt. An dieser Stelle hat sich das Team dazu entschieden nicht weiter mit der Datenbank zu arbeiten, da das Projektende näher kam. Es ist aber anzumerken, dass die Extraktion über eine URL aus der Datenbank möglich ist, da der Videoplayer mit der URL aus Kapitel 3 geladen wurde. Aber eine geeignete Seite für das Ablegen der Dateien muss gefunden werden. Aus diesem Grund wurde das Backend auch nicht implementiert.

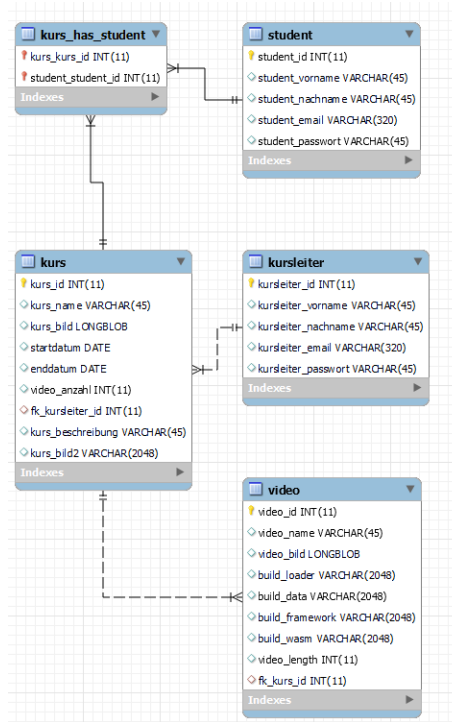


Figure 9: Erweitertes Entity Relationship Modell

5 Video Player

Der Video Player (im weiteren nur “Player” genannt) ist der Kern der E-Learning Plattform. Sowohl für die Erstellung neuer Inhalte als auch für das Konsumieren dieser Inhalte, ist der Player essentiell, um die 3D-Avatare und Animationen abzuspielen. Aus früherer Arbeit sind die Optionen zur Realisierung dieses Players reduziert worden auf WebGL, Unity und Unreal Engine. Dieses Kapitel beschreibt die Anforderungen an den Player, erläutert die Entscheidungen in Bezug auf diese Realisierung und geht ins Detail wie die Umsetzung funktioniert.

5.1 Ziele und Erwartungen

Der Player ist verantwortlich für die Darstellung der gewählten 3D-Avatare und das Abspielen der Animationsdaten eines gewählten Videos oder Motion Capture Clips. Neben den erwarteten Funktionen regulärer 2D Video Player, wie Starten und Pausieren des Videos und einer Timeline, um anzuzeigen wie weit das Video abgespielt wurde oder einen Timestamp zu wählen ab welchen das Video fortgesetzt werden soll, benötigt dieser Player mehrere 3D-spezifische und andere hilfreiche Funktionen.

Bezüglich regulärer Video Steuerung benötigt der Player eine Möglichkeit die Wiedergabegeschwindigkeit anzupassen, so dass der Benutzer die dargestellten Bewegungen in Slow Motion anschauen kann, um so ein besseres Verständnis für schnelle oder komplexe Bewegungsabläufe zu entwickeln. Zusätzlich sind weitere

Steuerungsmöglichkeiten erforderlich, um die verfügbare 3-dimensionale Szene zu nutzen. Speziell das Rotieren der Szene im 3-dimensionalen Raum, sowie rein und raus zoomen mit der Kamera. Mit diesen Erweiterungen der Steuerung, kann der Benutzer die Bewegung aus unterschiedlichen Winkeln betrachten und Details in der Bewegung besser untersuchen.

5.2 Auswahl des Frameworks

Die Auswahl des Frameworks für den Player bestimmt den weiteren Verlauf der Implementierung. Auch wenn von einer neutralen Perspektive alle drei Optionen (WebGL, Unity, Unreal Engine) gleichermaßen geeignet sind, haben alle Varianten ihre eigenen Vor- und Nachteile in Bezug auf die tatsächliche Umsetzung.

WebGL wurde vom Team früh im Entscheidungsprozess eliminiert, da keiner der Entwickler Erfahrung mit diesem Framework hat und die Natur als "Videospiel Engine" der beiden Alternativen bereits mehrere erforderliche Funktionen (das Darstellen von 3D-Modellen und das Abspielen von Animationen) von sich aus anbieten. Als Resultat wäre WebGL in der Entwicklung zu Zeitaufwändig, um eingesetzt zu werden. Sollte es zu einem späteren Zeitpunkt notwendig werden den Player zu optimieren, könnte WebGL erneut in Betracht gezogen werden, falls die beiden Videospiel Engines ungenutzten Ballast erzeugen, der die Performanz des Systems beeinträchtigt.

Der wichtigste Unterschied der beiden Engines ist ihre Implementation des WebGL exports. Mit Unreal Engine 5 wurde "Pixel Streaming" (Epic Games, Inc. 2022) eingeführt und Unterstützung für WebGL-Export beendet. Unity hingegen unterstützt WebGL-Export in der aktuellen Version (Ver 2021.3). Pixel Streaming verlagert die erforderliche Rechenleistung vom Browser des Clients auf einen Server. Die einzelnen Frames werden Serverseitig gerendert und anschließend zum Client gestreamt. Der WebGL-Export von Unity hingegen wird als Gesamtpaket an den Client übertragen und dort im Browser ausgeführt.

Um die Belastung der Server bei steigenden Benutzerzahlen so gering wie möglich zu halten, wird basierend auf diesen Informationen der Player mit Unity realisiert. Die gesteigerte Belastung des Endgeräts wird zum aktuellen Zeitpunkt als vernachlässigbar eingestuft. Bei steigender Komplexität des Players und der dargestellten Szenen wird empfohlen den genauen Einfluss dieser Entscheidung zu untersuchen und gegebenenfalls die Endgeräte zu entlasten, um die Einstiegsbarriere niedrig zu halten.

5.3 Umsetzung mit Unity

In der Unity-Szene, welche den Player darstellt, enthält eine Lichtquelle, eine Bodenebene, eine Kamera und einen 3D-Avatar. Um die in [5.2] angesprochene Rotation der Szene zu realisieren, wurde beschlossen nicht die gesamte Szene zu rotieren und stattdessen die Kamera um die Szene herum zu bewegen. Hierbei wurde ein existierender existierender Ansatz von Jasper Flick genutzt (Flick 2020). In seinem Projekt "Catlike Coding" beschreibt er Schritt für Schritt die Erstellung einer 3rd-Person Kamera, die einem bewegbaren Ball folgen und um diesen herum rotierbar sein soll. Diese "Orbit Camera" benötigt primär einen Fokus, welcher in diesem Fall als der 3D-Avatar gegeben ist. Die Kamera wird in einem zunächst fixen Standardabstand auf diesen Fokuspunkt ausgerichtet

und wird in der `LateUpdate()` Methode immer wieder erneut auf diesen Fokuspunkt ausgerichtet. Beim aktualisieren des Fokuspunkts, wird ebenfalls dafür gesorgt, dass die Kamera mit leichter Verzögerung und weichen Bewegungen dem Fokusobjekt folgt, um abrupte Bewegungen der Kamera zu vermeiden und *Motionsickness* zu reduzieren.

Neben der Aktualisierung des Fokuspunkts, können in der `LateUpdate()` Methode Benutzereingaben entgegen genommen werden durch den Aufruf der `ManualRotation()` Methode. Hierfür wurden Mausbewegungen auf die Input-Achsen "Vertical Camera" und "Horizontal Camera" gemapped. Diese Mausbewegungen werden dann in einen `Vector2` gespeichert und anschließend mit einer fixen Rotationsgeschwindigkeit skaliert. Damit Benutzer diese Kamerabewegung nur bewusst ausführen, wurde zusätzlich als Bedingung verknüpft, dass dieser Prozess nur ausgeführt wird, wenn der Benutzer die linke Maustaste drückt. Zuletzt wurde ebenfalls ergänzt, dass der bisher fixe Abstand der Kamera zum Fokuspunkt mit Hilfe des Mausekkrads verändert werden kann.

```

1 void ManualRotation()
2 {
3     if (Input.GetMouseButton(0))           //Left Mousebutton
4     {
5         Vector2 input = new Vector2(
6             Input.GetAxis("Vertical Camera"), //Mousemovement mapped to
7             Input.GetAxis("Horizontal Camera") //these Axis in Unity
8         );
9         const float e = 0.001f;
10        if (input.x < -e || input.x > e || input.y < -e || input.y > e)
11        {
12            orbitAngles+= rotationSpeed * Time.unscaledDeltaTime * input;
13        }
14    }
15
16    float scrollFactor = Input.GetAxis("Mouse ScrollWheel");
17
18    if (scrollFactor != 0)
19    {
20        distance = distance * (1f - scrollFactor);
21        if (distance > 20f)           //limits scroll distance
22            distance = 20f;           //to not go too far away
23        else if (distance < 1f)       //or too close
24            distance = 1f;
25    }
26 }

```

Listing 1: Methode `ManualRotation()` in `CatOrb`

Die Orbit Camera von Catlike Coding enthält noch weitere Funktionen, die speziell für das Steuern eines Balls durch einen Hindernissparkour relevant sind, aber für den Player keine Rolle spielen. Aus diesem Grund wurden diese nicht übernommen.

Animationen werden in Unity mit einem Animator abgespielt. In diesem werden die Animationen importiert und können manuell ausgewählt werden. Im aktuellen Stand des Players müssen neue Animationen per Hand eingefügt werden und das Projekt muss mit der neuen Animation separat exportiert werden, um mit dem System zu Funktionieren. Für spätere Revisionen des Projekts kann Versucht werden neue Animationen über einen Assetloader einzubinden und während der Laufzeit die korrekte Animation auszuwählen. Aus Zeitgründen wurde diese Animation für diesen Release jedoch Vernachlässigt. Ebenso wurde

damit experimentiert einen Idle Zustand zu implementieren, in welchen der 3D-Avatar vor und nach Animationen wechseln kann. Dieser bot allerdings keine Vorteile und wurde daher im Laufe des Projekts wieder entfernt. Jede weitere Kontrolle der Animationen wird über das Userinterface gesteuert.

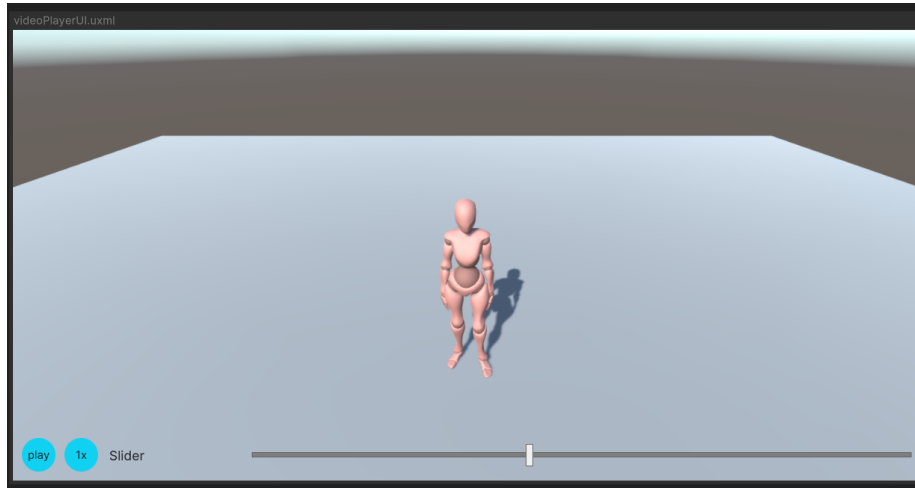


Figure 10: Screenshot aus dem Unity UI Builder. Zeigt den Ausgangszustand des User Interface für den Player.

Das in Figur 10 zu sehende Userinterface wird durch ein UIDocument Element realisiert und enthält zwei Buttons und einen Slider. In weiteren Revisionen könnte das Interface mit einer Erklärung der Steuerung ergänzt werden. Alternativ kann diese auf der Seite direkt eingeblendet werden. Die UIController Klasse kontrolliert alle Elemente des Interface. Um die Animation zu steuern wird dieser Klasse ebenfalls der Animator der Szene übergeben.

Die Klasse enthält zwei Click-Listener für die zwei Buttons. Der erste Button startet und pausiert die Wiedergabe der Animation indem die Timescale der Szene auf 0 oder 1 gesetzt wird. Der zweite Button verändert die Wiedergabegeschwindigkeit. Hierbei wird die Timescale weiter unterteilt in 1 für eine normale Wiedergabe, 0.5 für leichte Slowmotion und 0.25 um die Geschwindigkeit stark zu reduzieren. Diese Werte werden ebenfalls im ersten Button aufgegriffen, um beim abspielen der Animation die zuletzt gewählte Wiedergabegeschwindigkeit zu nutzen.

```

1  void PlayButtonPressed()
2  {
3      switch (playing)
4      {
5          case true:
6              Time.timeScale = 0;           //stop playback
7              playing = false;
8              playButton.text = "play";
9              break;
10         case false:
11             Time.timeScale = playbackSpeed; //resume playback
12             playing = true;
13             playButton.text = "pause";
14             break;

```

```
15     }  
16 }
```

Listing 2: Methode zum starten/stoppen der Wiedergabe

Der Slider stellt die Timeline der Animation dar. Hierbei wird über den Animator die Länge und der aktuelle Zeitpunkt der Animation abgerufen, um für den Maximalwert und die aktuelle Position des Sliders genutzt zu werden. Zusätzlich werden beide Werte in Minuten und Sekunden umgerechnet, welche im Textfeld des Sliders dargestellt werden. In der Update() Methode wird zudem abgefragt, ob die Position des Sliders vom Benutzer verändert wurde und setzt in dem Fall die Animation auf den entsprechenden Zeitpunkt.

5.4 Evaluation und Potential

Im aktuellen Zustand verfügt der Player über alle geplanten Steuerungsoptionen. Als weitere Verbesserung bietet sich an die Rotation der Kamera zu limitieren, um zu verhindern dass Benutzer diese unter die Bodenebene bewegen. Ebenfalls ist es denkbar eine weitere Bewegungsmöglichkeit zu ergänzen mit welcher der Benutzer den Fokuspunkt der Kamera manuell verändern kann, um noch besser auf einzelne Elemente der Bewegung zu fokussieren.

Bezüglich der 3D-Avatare und Animationen ist der Player derzeit sehr limitiert. Neue Avatare und Animationen müssen manuelle ergänzt werden und als kompletter neuer Export in den Rest des Systems eingespeichert werden. Diese Limitierung muss gelöst werden, bevor das System massentauglich sein kann.

Insgesamt zeigt sich Unity jedoch sehr geeignet für die Erstellung des Players und bietet Raum für Erweiterungen und Verbesserungen in weiteren Iterationen des Projekts.

6 Motion Capture

Die geplanten Aufnahmen vom Paartanz mit dem Motion Capture System Capture Live mussten an das Ende des Semesters verschoben werden, da das Capture Live System technische Probleme hatte. Die Entwicklungsphase wurde so angepasst, dass keine selbst erstellten Aufnahmen nötig sind und stattdessen mit frei verfügbaren Motion Capture Daten aus dem Internet gearbeitet wird. Am Ende des Semesters sind Motion Capture Aufnahmen von Paartänzen entstanden. So wurden die Tänze Discofox, Tango und langsamer Walzer aufgenommen. Das Capture Live ermöglicht das Tracken von zwei Personen gleichzeitig (vergleiche Figure 11), jedoch sind die Aufnahmen nicht einwandfrei. Dies liegt sowohl an der neuen Kalibrierung, da nun acht statt zwölf Kameras verwendet werden, als auch an anderen Faktoren wie der aufgenommenen Person und Kleidungsfarbe oder -stücke. Die weibliche Person, die enge rote und blaue Kleidung trägt, wird vom System problemlos erkannt und getrackt. Die männliche Person mit lockerer hellgrauer und schwarzer Kleidung wird vom System erst nach mehreren Versuchen erkannt. Beim Tracken des Paartanzes werden die Gliedmaßen (vor allem die Beine) der männlichen Person bei Bewegungsabläufen größtenteils nicht richtig erkannt und es werden fehlerhafte Daten ausgegeben (siehe Figure 12).



Figure 11: Bewegungsaufnahme im Paartanz

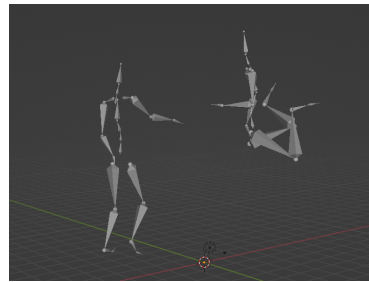


Figure 12: Fehlerhafte Bewegungsaufnahmen im Paartanz

Dies unterscheidet sich nicht am Tanzstil oder -geschwindigkeit, daher hängt die Qualität der Motion Capture Daten stark von der aufgenommenen Person, Kleidung und gegebenenfalls weiteren Faktoren wie der Bodenfarbe, die mit der Kleidungsfarbe identisch ist ab. Kleine Fehlstellungen an einzelnen Gliedmaßen wie einzelne Hände oder Füße können in Blender nachbearbeitet werden. Hierzu wird ein Knochen des Modell Skeletts ausgewählt und positioniert, sodass die neue Position in der Playback Spur mithilfe von Keyframes gespeichert werden kann. Da jedoch die Anzahl der fehlerhaften Bewegungen so hoch ist und mehrere Knochen manuell neu positioniert werden müssen, gehen dadurch die natürlichen Bewegungen verloren.

Es werden jeweils ein männlicher und ein weiblicher Avatar¹ aus der Seite free3d verwendet, welche die Tanzbewegungen darstellen sollen (siehe Figure 13). Diese Modelle beinhalten schon Animationen, die entfernt werden müssen, sodass nur

¹Die Avatare sind für den persönlichen Gebrauch gedacht und können ohne Zustimmung des Rechtsinhabers verwendet werden. Sollte in der Zukunft das Projekt den Einsatz im Lehrbereich weiterentwickelt werden, müssen Alternativen gesucht werden.



Figure 13: Paartanz mit 3D Avataren

ohne Skelettmodell bestehen bleibt. Das Rigging der Tanzaufnahme an den Avatar wird mit Blender manuell ausgeführt, indem der Avatar an das Skelett mit den Bewegungen verankert wird.

Dieser Schritt wird für den männlichen und weiblichen Avatar separat ausgeführt. Die Avatare werden bei bestimmter Bewegungsausführung nicht realistisch sondern mit verdrehten oder eckigen Armen abgebildet. Zudem wird bei der Bewegungsdarstellung über die Avatare deutlich, dass es Probleme bei Überlappungen wie der Hand auf der Schulter oder Fuß auf Fuß gibt. Da die Aufnahmen am Ende des Projekts entstanden sind, konnten diese Probleme nicht rechtzeitig behoben werden. Dennoch werden die Tanzbewegungen gut über die Avatare präsentiert. So können in der nächsten Projektphase die Aufnahmen ausgebessert werden. Eine Nachbearbeitung einzelner Gliedmaßen wäre zu aufwendig.

7 Minimum Viable Product

In diesem Semester wurde ein Minimum Viable Product (MVP) entwickelt, welches die wichtigsten Funktionen des Projekts aus der Konzeptionsphase umfassen soll. Ursprünglich wurde im letzten Semester eine Entwicklung für die Lehrkräfte und Lernende angestrebt. Diese umfasst einen Videoeditor für Lehrkräfte (Editor MVP), in dem Motion Capture Bewegungen über Avatare in Clips dargestellt und mithilfe von Texten und Audio zu einem Video zusammengeschnitten werden. In einem interaktiven Videoplayer können Lernende (KursUser MVP) diese Videos, die in einem Kurs hinterlegt werden, betrachten.

Am Anfang der Entwicklungsphase wurden die funktionalen Anforderungen aus der letzten Projektphase priorisiert und anhand dieser neue User Stories erstellt und kategorisiert. So wurden Anforderungen, die für Lehrer und Schüler Interaktionen mit und innerhalb von Kursen beinhalten, für das MVP weggelassen. Der Fokus sollte somit in der Erstellung und dem Betrachten von interaktiven Videos liegen. Hierzu wurden bestehende User Stories ergänzt und in drei Kategorien eingeteilt. Die erste Kategorie enthält die wichtigsten User Stories, die Aktivitäten mit den Motion Capture Aufnahmen beschreiben. Die zweite Kategorie beinhaltet wichtige User Stories, die Hilfestellungen wie Textinhalte für Motion Capture Aufnahmen in Videos beinhalten. In der letzten Kategorie sind die für das MVP aussortierten User Stories enthalten, die Kursaktivitäten berücksichtigen.

Nachdem die Vorbereitung abgeschlossen war und die Entwicklung des Video Players, Frontends, Backends und der Datenbank angefangen hat, haben wir jedoch gemerkt, dass die Umsetzung des geplanten MVPs mit der Unterteilung in Lehrkräfte und Lernende nicht möglich ist. Die Funktion eines Videoeditors stellt dabei das größte Problem dar. Wenn eine Lehrkraft ein Video erstellen möchte, müsste eine Bewegungsaufnahme in der Kameraposition und Zoomstärke gespeichert werden, wie sie die Lehrkraft im Videoeditor anpasst. Dazu muss es möglich sein über das Web die Daten im Unityprojekt zu bearbeiten und abzuspeichern, sodass die festgelegte Kameraposition Lernenden im Videoplayer angezeigt wird. Die Umsetzung ist jedoch ohne Vorkenntnisse in Unity in einem Semester nicht möglich. An dieser Stelle wurde beschlossen das MVP aus Sicht der Lehrkräfte nicht zu entwickeln, da der Aufwand zu groß ist. Zudem konnten die Unity-Build Dateien nicht aus der SQL-Datenbank in Vue angemessen extrahiert werden. Aus diesem Grund wurde entschieden, die Datenbank und ein Backend für das MVP nicht zu verwenden, da die Daten im Code hinterlegt werden können. Stattdessen wird das KursUser MVP mit Fokus auf einen interaktiven Videoplayer erstellt. Aus diesem Grund werden die zuvor festgelegten User Stories nicht entwickelt, da sie auf der Grundlage des Videoeditors erstellt wurden.

Für das MVP stellt die Kursübersicht von Lernenden die Startseite dar (14). Dort sind alle Kurse gelistet, denen ein Lernender beigetreten ist. Die Kurse werden über Cards präsentiert und können über Buttons geöffnet werden. Im MVP werden nicht anklickbare Kurse mit einem deaktivierten Button versehen, um den Nutzer durch das MVP zu navigieren. Da für die Entwicklung keine Datenbank verwendet wurde, kann auch kein Fortschritt zu einzelnen Kursen gespeichert werden. In einem weiteren Screen werden die in einem Kurs angebotenen Videos über Cards gelistet (15). Beim Anklicken eines Videos gelangt man auf einen Screen mit dem interaktiven Videoplayer (16) (17). Dort hat der

Nutzer die Möglichkeit, durch die interaktive 3D Ansicht des Videos zu steuern und die Bewegungen verlangsamt anzusehen.

8 Review

Die eingesetzte Ziele dieser Arbeit waren nicht einfach und zeitlich wie geplant erreicht. Die Aufnahmen vom Paartanz waren am Anfang wegen verschiedenen System Probleme nicht realisierbar. was die geplante reihenfolge der Arbeit ändert hat. Dies wurde Ende des Semesters gelöst und teilweise fehlerfreie Tanzaufnahmen wurden aufgenommen. Aber es gaben immer kleinigkeiten die bei der Bewegungen fehlen wie zum Beispiel kleine Handbewegungen.

Auf einer anderen Ebene entstanden paar Probleme mit der Umsetzung von Videoplayer mit der Unity. Einerseits erfordert jede Animation sowie jeder Avatar einen eigenen kompletten Export. Dies hat zur Folge, dass es extra Arbeit folgte und somit es wurde mehr Zeit benötigt. Andererseits war eine automatische Auswahl der richtigen Animation nicht machbar. Dies war zwar technisch möglich aber durch das fehlendes Wissen der Teammitglieder sowie die entstandene Zeit Probleme wurden diese Probleme nicht gelöst.

9 Fazit

Das Ziel für dieses Projekt war das Erstellen eines Minimum Viable Products (MVP) für die, in einem vorangegangenen Projekt konzipierten, e-Learning Plattform mit Motion Capture. Das MVP sollte ursprünglich den Kursleiter-spezifischen Bereich umfassen, in welchem diese Lehrvideos erstellen können.

Vor allem in den Anfangsphasen des Projekts führten Vorkenntnisse des Teams zu mehreren Verzögerungen und Überarbeitungen des Zeitplans. Während das Team neue Fähigkeiten parallel zum Projekt entwickeln musste und dadurch der geplante Zeitabschnitt für die Umsetzung des MVPs kürzer war als von Anfang an angenommen, wurde der Plan für den MVP geändert und umfasste nun stattdessen den Bereich in welchem Kursteilnehmer bereits erstellte Lehrvideos anschauen können.

Obwohl der Zeitplan des Projekts wiederholt durch fehlende Fähigkeiten des Teams verzögert wurde, konnte der geänderte Plan für den MVP zufriedenstellend innerhalb der Frist umgesetzt werden. Mit Unterstützung der Betreuer und zusätzlichem Zeitaufwand zum recherchieren der problematischen Elemente während der Implementierung, konnten die meisten Probleme erfolgreich vom Team gelöst werden. Nur die auf externe Probleme zurück zu führenden Schwierigkeiten beim aufnehmen von Beispiel Clips im Motion Capture Studio, mussten im Projekt durch Verwenden von existierenden Animationen umgangen werden.

Für zukünftige Projekte sollte festgehalten werden, dass zu Beginn des Projekts ein besseres Verständnis für die Fähigkeiten des Teams erforderlich ist, um den Ablauf des Projekts besser abzuschätzen und zu planen. Das Projekt konnte erfolgreich innerhalb der Frist durchgeführt werden, jedoch hätte bessere Vorbereitung in diesem Aspekt voraussichtlich zu einem zufriedenstellenderem Ergebnis geführt.

List of Figures

1	Das erste Domänenmodell für die e-Learning Plattform mit Motion Capture. Stellt die ersten Geschäftsobjekte und ihre Beziehungen dar.	4
2	Modell für die Seitenstruktur der webseite	5
3	Styleguide Version 1	6
4	Styleguide Version 2	7
5	Architektur Skizze. Zeigt die wesentlichen Elemente des e-Learning Systems unterteilt (von oben nach unten) in Geschäfts-, Applikations- und Technologie-Ebene.	9
6	Modifikation der gesamten Architektur Skizze. Der KursUser MVP (links) enthält KursUser GUI und Applikationskomponente, der Editor MVP (rechts) enthält Editor GUI und Applikationskomponente. Beide MVP enthalten eine gesamte Datenbank und die Video Player Komponente.	10
7	Die Home-Seite der erstellten Webseite	11
8	Die überarbeitete Home-Seite der erstellten Webseite	12
9	Erweitertes Entity Relationship Modell	16
10	Screenshot aus dem Unity UI Builder. Zeigt den Ausgangszustand des User Interface für den Player.	19
11	Bewegungsaufnahme im Paartanz	21
12	Fehlerhafte Bewegungsaufnahmen im Paartanz	21
13	Paartanz mit 3D Avataren	22
14	Screen: Kursübersicht	30
15	Screen: Videoübersicht	30
16	Screen: Macarena Video	31
17	Screen: YMCA Video	31

Listings

1	Methode ManualRotation() in CatOrb	18
2	Methode zum starten/stoppen der Wiedergabe	19

Literaturverzeichnis

Epic Games, Inc. 2022

EPIC GAMES, INC.: *Pixel Streaming*. <https://docs.unrealengine.com/5.0/en-US/pixel-streaming-in-unreal-engine/>. Version: 2022. – Last accessed 22 September 2022

Flick 2020

FLICK, Jasper: *Catlike Coding - Orbit Camera*. <https://catlikecoding.com/unity/tutorials/movement/orbit-camera/>. Version: Juli 2020. – Last accessed 22 September 2022

Google 2022

GOOGLE: *Material Design*. <https://material.io/>. Version: 2022. – Last accessed 14 September 2022

Lennon 2009

Kapitel 4. In: LENNON, Joe: *Developing Applications with CouchDB*. Apress, Berkeley, CA, 2009. – ISBN 978-1-4302-7236-6, 185–209

Subramanian 2019

SUBRAMANIAN, Vasan: *MongoDB*. https://doi.org/10.1007/978-1-4842-4391-6_6. Version: 2019. – Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node

A Anhang

A.1 MVP

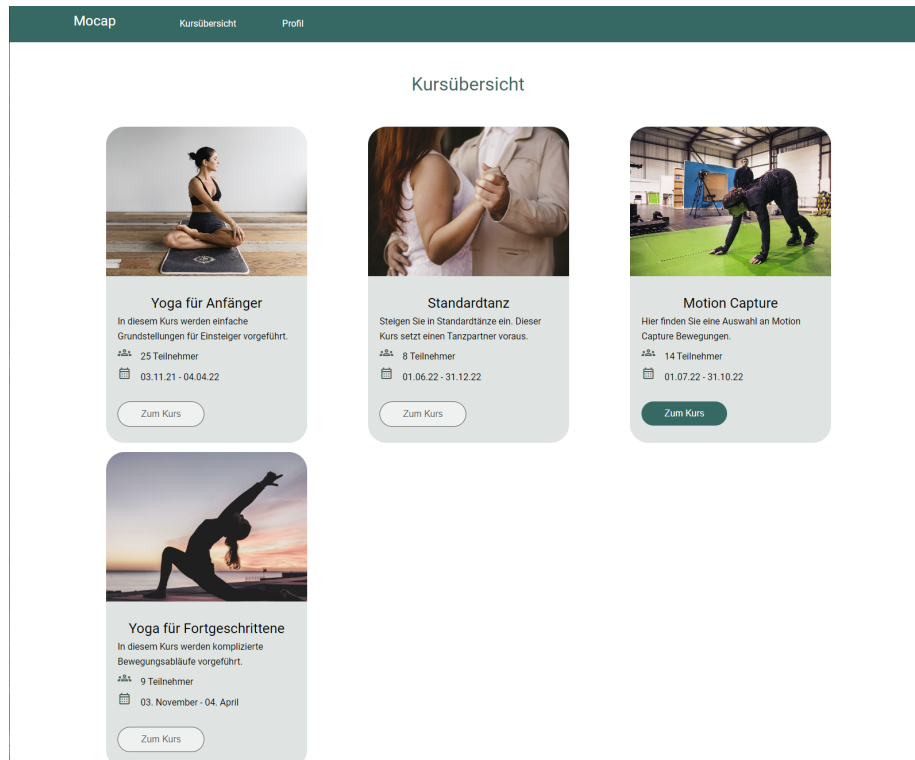


Figure 14: Screen: Kursübersicht



Figure 15: Screen: Videoübersicht

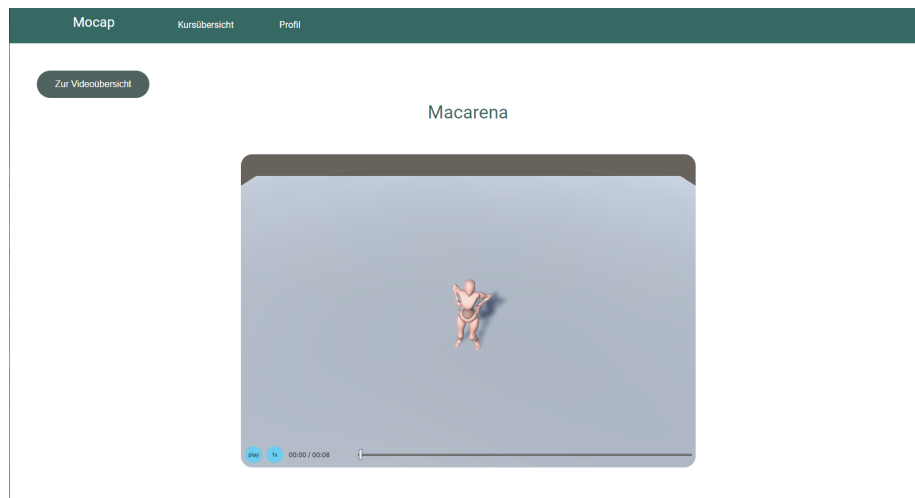


Figure 16: Screen: Macarena Video

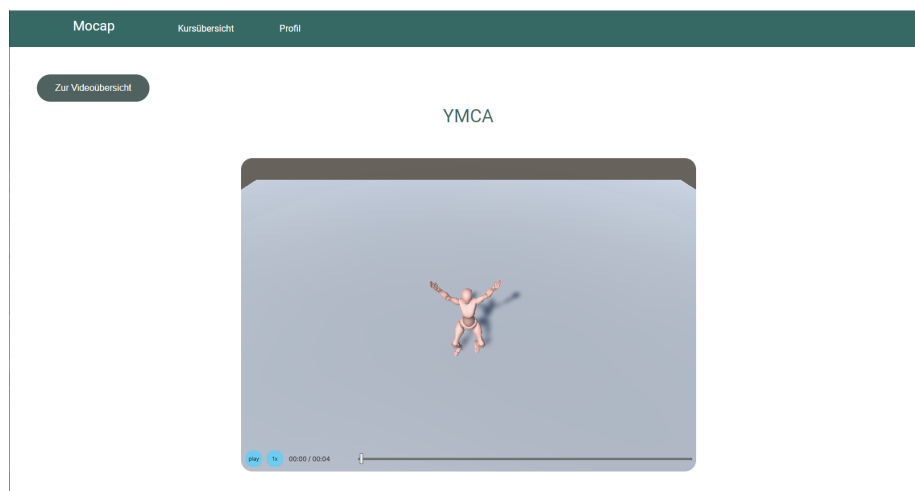


Figure 17: Screen: YMCA Video