



Hochschule Karlsruhe  
Technik und Wirtschaft  
UNIVERSITY OF APPLIED SCIENCES

# Bachelor-Thesis

„Entwicklung und Konzeption eines E-Learnings mit Lernvideos zu Virtual-Reality- und Augmented-Reality-Anwendungen auf Basis von Vorlesungsunterlagen“

von

Julia Heidenreich (57048)

**Betreuer:** Herr Prof. Dipl.-Ing. Martin Schober

**Zweitgutachter:** Prof. Dipl.-Inf. Sissi Elisabeth Closs

**Bearbeitungszeitraum:** 01.12.2020 – 30.04.2021

# Inhaltsverzeichnis

<b>1. Einleitung</b> .....	<b>5</b>
1.1. Projektbeschreibung .....	5
1.2. Ausgangssituation.....	5
1.3. Aufbau der Arbeit.....	5
<b>2. Warum E-Learning?</b> .....	<b>7</b>
<b>3. Wissensvermittlung</b> .....	<b>8</b>
3.1. Wissen vermitteln.....	8
Ein Ziel setzen.....	8
Das Problem mit der Zeit .....	8
Probieren geht über Studieren.....	9
Die Motivation steigern.....	9
3.2. E-Learnings aufbauen.....	10
Die richtige Projektauflösung ermitteln .....	10
Displayauflösung & Zielgeräte .....	10
Zielplattform & Publikationsformat .....	11
3.3. Softwarewissen umsetzen .....	11
Die Übungsaufgabe „üben“ .....	11
<b>4. Praktische Ausarbeitung</b> .....	<b>12</b>
4.1. Definition der Zielgruppe.....	12
Anforderungen an das E-Learning hinsichtlich der Zielgruppe .....	12
4.2. Ist-Analyse der Vorlesungsunterlagen.....	13
Unity Einstieg.....	13
Unity + C#: Texte eingeben/ausgeben durch Mausclick .....	17
Unity Collision Detection .....	20
Zusammenfassung.....	24
Unity Einstieg.....	24
Unity + C#: Texte eingeben/ausgeben durch Mausclick .....	24

Unity + C#: Klicks auf 3D-Objekte zählen und ausgeben.....	25
4.3. Soll-Konzept für das geplante E-Learning.....	25
Geplanter Aufbau .....	26
Unity Einstieg.....	26
Unity + C#: Texte eingeben/ausgeben durch Mausklick .....	28
Geplante Technologien zur Umsetzung .....	28
4.4. Umsetzung des Soll-Konzepts.....	29
Erstellen von Lernvideos per Screencast .....	29
Was ist ein Screencast?.....	29
Worauf muss man achten? .....	29
Screencast mit Adobe Captivate .....	31
Screencast mit GeForce Experience .....	37
Praktische Durchführung: Erstellung der Lernvideos für das geplante E-Learning .....	38
Videos mit Adobe Premiere Pro nachbearbeiten und schneiden .....	40
E-Learning mit Adobe Captivate erstellen .....	46
Ein neues Projekt anlegen .....	46
Die Designfarbe ändern .....	47
Textfelder einfügen.....	49
Ein Folienvideo einfügen .....	50
Anzeigedauer der Elemente in der Zeitleiste festlegen .....	50
Weitere Folien erstellen.....	52
Das Projekt veröffentlichen .....	53
5. Leitfaden für das Erstellen eines E-Learnings auf Basis von Vorlesungsunterlagen .....	55
Ziel .....	55
Benötigte Software .....	55
1. Lernvideos per Screencast erstellen .....	55
2. Erstellte Videos zuschneiden und bearbeiten .....	56
3. E-Learning erstellen .....	57



6. Abbildungsverzeichnis.....	59
7. Quellenverzeichnis .....	61
Bücher .....	61
Vorlesungsunterlagen .....	61
Internet .....	62
8. Eidesstattliche Erklärung .....	63

# 1. Einleitung

## 1.1. Projektbeschreibung

Zu den Übungsaufgaben aus den Unity-Vorlesungsunterlagen sollen E-Learnings mit Lernvideos erstellt werden. Diese Lernvideos sollen Unity-Anwendern den Einstieg in das Programm erleichtern. Zielgruppe hierbei sind die Studierenden der HsKA, die zwar ein wenig JavaScript- und CAD-Erfahrung haben, aber das Programm ansonsten nicht kennen. Durch die praktische Umsetzung der Übungsaufgaben für die Lernvideos fallen Fehler, Lücken und sonstige Stolperstellen in den Vorlesungsunterlagen automatisch auf. Diese können dann direkt verbessert bzw. ergänzt und die Vorlesungsunterlagen somit optimiert werden.

Aus den erstellten Lernvideos und den Vorlesungsunterlagen sollen E-Learnings erstellt werden. Diese E-Learnings sollen den Studierenden kurz aufzeigen, was Unity ist, wofür es verwendet wird und einen Kontext zu den bestehenden Übungsaufgaben herstellen. Die Inhalte der bereits bestehenden Vorlesungsunterlagen werden also um eine Einleitung und um Kontexte zu den Übungen erweitert, die den Nutzen bzw. den Sinn der jeweiligen Übung erläutern sollen.

Des Weiteren soll ein Leitfaden zur Erstellung weiterer Lernvideos erstellt werden. Dieser Leitfaden soll Assistenten und Hiwis bei der Umsetzung von weiteren Lernvideos helfen.

## 1.2. Ausgangssituation

Aktuell befinden sich alle Vorlesungsunterlagen zu Unity in Form von mehreren PDF-Dateien auf der Webseite [www.technischeredaktion.com](http://www.technischeredaktion.com). Dieses PDF-Dateien beinhaltet mehrere Übungsaufgaben, die den Studierenden die verschiedenen Anwendungsgebiete und Umsetzungsmöglichkeiten von Unity zeigen sollen. Diesen Übungsaufgaben fehlt derzeit allerdings noch der Kontext. Man steigt direkt und ohne jegliches Vorwissen mit den Übungen in das Programm ein.

## 1.3. Aufbau der Arbeit

Diese Arbeit ist in 3 große Hauptbereiche aufgeteilt. Im ersten Teil wird es um die **Definition von E-Learnings** gehen. Hier wird also erläutert, was E-Learnings überhaupt sind und warum diese verwendet werden sollen, um den Studierenden den Umgang mit Unity beizubringen.

Im zweiten Teil wird dann **Wissensvermittlung im Allgemeinen** behandelt. Hier wird genauer erläutert, wie man Wissen am effektivsten vermitteln kann, wie E-Learnings aufgebaut werden und wie man Softwarewissen am besten umsetzen und das Wissen weitergeben kann.

Im letzten Teil folgt dann die Dokumentation zur **praktischen Umsetzung des E-Learnings**. Hier wird zunächst die Zielgruppe definiert. Anschließend wird der Ist-Zustand der Vorlesungsunterlagen dargestellt, anhand dessen dann ein Soll-Konzept für die geplanten E-Learnings erstellt wird.

Darauf folgt die Umsetzung des Soll-Konzepts. Hier wird darauf eingegangen, wie Lernvideos per Screencast erstellt werden können, wie diese Lernvideos anschließend bearbeitet werden können und welche Software hierfür zur Verfügung steht. Darauf folgt dann auch direkt eine Dokumentation zur praktischen Umsetzung, in der unter anderem gezeigt wird, welche Probleme bei der Umsetzung aufgetaucht sind und wie diese gelöst wurden.

Zuletzt wird gezeigt, wie die E-Learnings dann letztendlich erstellt wurden und worauf hierbei geachtet werden musste.

## 2. Warum E-Learning?

Warum ist ein E-Learning sinnvoll, um den Studierenden den Umgang mit Unity beizubringen? Und was ist überhaupt ein E-Learning?

*„Unter dem Begriff E-Learning (electronic learning) versteht man das Lernen unter Verwendung von elektronischen Geräten und Medien.“ E-Learning-Plattformen (o. J.): „Was ist E-Learning?“ <<https://www.e-learning-plattformen.de/was-ist-e-learning-definition/>> [Stand: 2021]*

Bei diesen elektronischen Geräten kann es sich beispielsweise um Smartphones, Tablets, Computer und Laptops handeln.

E-Learnings können von überall aus aufgerufen werden. Wenn das E-Learning offline nicht zur Verfügung stehen sollte, dann benötigt man zusätzlich zu dem elektronischen Gerät auch einen Internetzugang. Die Lerninhalte, die im E-Learning übermittelt werden sollen, werden digital zur Verfügung gestellt. Wenn man sich beispielsweise auf YouTube Lernvideos oder Tutorials ansieht, so handelt es sich dabei auch um E-Learnings.

Warum ist es sinnvoll, den Studierenden die Unity-Vorlesungsunterlagen in Form von E-Learnings zur Verfügung zu stellen?

Unity ist eine 3D-Entwicklungsplattform für Spiele, Animationen, Filme, u.v.m. Für Anwender wie unsere Studierenden, die in der Regel noch nie mit diesem Programm gearbeitet haben, kann die Systemoberfläche zu Anfang sehr verwirrend und umfangreich sein. Ein E-Learning mit Lernvideos bietet die Möglichkeit, im Programm direkt 1:1 das nachzumachen, was im Lernvideo gezeigt wird. Der Anwender muss also nicht erst eine ggf. komplizierte Schrittanleitung nachvollziehen, sondern kann mit eigenen Augen sehen, was getan werden muss und die Schritte zeitgleich nachmachen. Und das Beste: Die Studierenden können die Lernvideos jederzeit pausieren, zurückspulen und von vorne starten, so oft sie möchten.

Außerdem haben die Studierenden in den Poolräumen Zugriff auf PCs mit Internetzugang, womit die Hardware-Anforderungen mehr als erfüllt sind.

Ein weiterer Vorteil ist, dass die Studierenden schon vor dem Durchführen der Übung sehen können, was das Ergebnis der Übung sein wird und was sie in dieser Übung sozusagen lernen können. Dafür muss direkt zu Beginn des E-Learnings natürlich eine Folie eingefügt werden, in der das Ziel der Übung und ein Demovideo zu sehen sind.

## 3. Wissensvermittlung

In diesem Kapitel wird erklärt, wie man Wissen am besten vermitteln kann und welche Anforderungen an ein gutes E-Learning gestellt werden müssen.

Denn um ein gutes E-Learning erstellen zu können, muss man wissen, wie man an andere Personen effektiv Wissen vermitteln kann. Dafür muss man wissen, wie man E-Learnings richtig aufbaut und wie man Softwarewissen umsetzt. Soll das E-Learning außerdem auch Lernvideos beinhalten, so muss man vorab auch wissen, welche Eigenschaften ein gutes Lernvideo haben muss und wie diese erstellt werden.

### 3.1. Wissen vermitteln

#### Ein Ziel setzen

Bevor man Wissen vermitteln kann, muss man sich zunächst darüber im Klaren sein, was überhaupt das Ziel dieser Wissensvermittlung ist. Will man den teilnehmenden Personen nur einen „Überblick“ über das Thema vermitteln? Oder will man ihnen wirklich etwas beibringen? Vielleicht sogar mit dem Ziel, dass sie im Anschluss an die Lerneinheit das erlangte Wissen in der Praxis umsetzen können?

Im Fall dieser Arbeit, sollen den Studierenden anhand von Übungen verschiedene Unity-Anwendungsszenarien gezeigt und beigebracht werden, sodass sie diese Szenarien im Anschluss selbstständig umsetzen können. Den Studierenden soll beispielsweise beigebracht werden, wie sie in Unity ein Auto in einer 3D-Welt erstellen können, und wie man dieses Auto so programmieren kann, dass man es per Tastatur innerhalb dieser 3D-Welt fahren und steuern kann.

Wie lässt sich dieses Ziel also am schnellsten erreichen?

#### Das Problem mit der Zeit

Die größte Herausforderung bei der Vermittlung von Wissen aller Art ist und bleibt die Zeit. Den Lernenden soll in möglichst kurzer Zeit möglichst viel Wissen nahegebracht werden und das so, dass die Lernenden im Anschluss an die Lerneinheit auch ohne fremde Hilfe das Gelernte umsetzen können.

Mit einer kurzen Power-Point-Präsentation wird sich dieses Ziel nicht erreichen lassen. Die Studierenden müssen sich intensiv mit dem Thema auseinandersetzen, dürfen gleichzeitig aber nicht überfordert werden. Das Wissen, das vermittelt werden soll, darf nicht zu umfangreich behandelt werden. Gleichzeitig darf das Thema aber auch nicht zu schnell abgehandelt werden, da den Lernenden sonst essenzielle Zusammenhänge vorenthalten werden und die Lerneinheit letztendlich mehr Fragen als Antworten aufwirft.

Übertragen auf diese Arbeit darf ein gutes E-Learning also weder zu lang noch zu kurz sein. Aber wie findet man den idealen Umfang für das E-Learning heraus? Woher weiß ich, ob mein E-Learning zu umfangreich oder zu knapp behandelt ist?

## Probieren geht über Studieren

Die Perfekte Formel für die Länge einer Lerneinheit – oder wie in diesem Fall die Länge eines E-Learnings – gibt es wahrscheinlich nicht. Die beste Möglichkeit, um den idealen Umfang herauszufinden, ist es einfach auszuprobieren. Man erstellt das E-Learning mit all den Inhalten, die man für Wichtig hält, und schaut sich dann das Resultat an. Schon beim ersten Selbsttest merkt man, wie die erstellte Lerneinheit einem selbst gefällt.

Dauert die Lerneinheit zu lang und man hat das Gefühl, es nimmt kein Ende?

Dann sollte man versuchen, sich kürzer zu fassen. Falls dadurch jedoch wichtige Zusammenhänge verloren gehen könnten, dann kann man auch versuchen, die Lerneinheit in zwei Lerneinheiten aufzuteilen und ein Zwischenergebnis zu ergänzen.

Ist die Lerneinheit zu kurz und man fragt sich im Anschluss, was man dadurch überhaupt gelernt hat?

Dann sollte man sich nochmal genau überlegen, was überhaupt das Ziel dieser Lerneinheit war und ob das nötige Wissen, um dieses Ziel zu erreichen, vermittelt wurde.

Das soll aber nicht bedeuten, dass ein kurzes E-Learning schlecht ist! Solange das Ziel erreicht wird, gilt je kürzer desto besser. Je schneller man lernt, wie etwas funktioniert, desto motivierter ist man auch, weiter zu lernen. Daher bietet es sich an, eine große Lerneinheit in mehrere kleine Zwischeneinheiten aufzuteilen. Am Ende jeder Zwischeneinheit kann man dann sagen, dass man nun gelernt hat, wie man ein „xy“ erstellt oder wie man „xy“ anwendet etc.

## Die Motivation steigern

Jede erfolgreich absolvierte Einheit stellt für den Lernenden ein Erfolgserlebnis dar und motiviert ihn, weiterzumachen. Eine große Lerneinheit hingegen, ohne die gerade erwähnten Zwischenergebnisse, könnte sich durchaus negativ auf die Motivation auswirken und birgt die Gefahr, dass die Lerneinheit frühzeitig abgebrochen und das Ziel somit nicht erreicht wird.

Auf diese Arbeit bezogen wäre der Idealfall also, dass die Lerneinheit das Interesse der Studierenden weckt und sie aus eigenem Antrieb den Umgang mit Unity lernen möchten und es nicht nur tun, weil sie es (Lehrplanbedingt) tun müssen.

Die Motivation wird aber nicht nur durch Zwischenergebnisse gesteigert. Auch ein gelungener kurzer prägnanter Einstieg ist essenziell. Den Studierenden muss direkt klar ersichtlich sein, was ihnen in den folgenden Lerneinheiten beigebracht wird und vor allem, was sie mit dem neu gewonnenen Wissen überhaupt anfangen können. Die Motivation, etwas Neues zu lernen, ist viel größer, wenn man weiß, WIESO man es überhaupt lernen soll und was man selbst davon hat bzw. was man mit dem neu gewonnenen Wissen anfangen kann.

## 3.2. E-Learnings aufbauen

Wenn man ein gutes E-Learning erstellen möchte, dann sollte man zuallererst wissen, wie so ein E-Learning überhaupt aufgebaut wird.

*„Am Anfang aller Überlegungen steht die angemessene Konzeption des Projektes.“ (Uhrig 2012:25)*

### Die richtige Projektauflösung ermitteln

Vor dem Projektstart sollte man sich gut überlegen, welche Auflösung das geplante Projekt haben soll. Denn sowohl eine zu hohe als auch eine zu niedrige Auflösung kann die Qualität des E-Learnings verschlechtern. Am besten ist es, wenn das Projekt ohne Skalierung weitergegeben werden kann. Denn gerade bei Softwareaufnahmen leidet sehr schnell die Lesbarkeit sämtlicher Oberflächentexte, sobald diese skaliert werden. Außerdem könnte eine Skalierung dazu führen, dass aufgrund von Platzmangel im Browser Scrollbalken angezeigt werden – was wiederum dazu führen könnte, dass die Wiedergabeleiste oder das Inhaltsverzeichnis nicht mehr direkt sichtbar sind (vgl. Uhrig 2012:26).

### Displayauflösung & Zielgeräte

Wenn man nicht genau weiß, welche Displayauflösung die Zielgeräte des Projekts haben, dann muss man mit einer Annäherung an die richtige Displayauflösung arbeiten. Aktuell gängige Auflösungen sind:

Zielgerät	Kleinste gängige Auflösung	Verhältnis
Notebook	1366 x 768 px	16:9
	1280 x 800 px	16:10
Tablet-PC	800 x 600 px	4:3
	800 x 480 px	16:9
Smartphone	480 : 320 px	3:2

*Tabelle 1: Kleinste gängige Auflösung nach Zielgeräten. (Uhrig 2012:27)*

## Zielformat & Publikationsformat

Neben dem Zielgerät wirken sich auch die Zielformat und das Publikationsformat stark auf die Projektgröße und den Anzeigebereich aus.

Im Falle dieser Arbeit wird es sich jedoch um ein Projekt im MP4-Format handeln, wodurch die Zielformat und das Publikationsformat keine Rolle spielen werden. MP4-Dateien werden nämlich in einem Player mit Vollbildfunktion und einer vom Player bereitgestellten Wiedergabeleiste abgespielt. Diese Wiedergabeleiste wird im Vollbildmodus automatisch ausgeblendet, sodass die Displaygröße 1:1 genutzt werden kann. (vgl. Uhrig 2012:27)

### 3.3. Softwarewissen umsetzen

Wie in der Einleitung bereits erwähnt, soll im Rahmen dieser Arbeit ein E-Learning zu Übungsaufgaben für das Programm Unity erstellt werden. Dieses E-Learning wird im Wesentlichen Schritt-für-Schritt-Anleitungen in Form von Lernvideos beinhalten. Bei diesen Lernvideos wird es sich um Bildschirmaufnahmen handeln, in denen die Übungsaufgaben in Unity durchgeführt werden. Um diese Bildschirmaufnahmen zu erstellen, bedarf es zunächst einiger Vorbereitung.

#### Die Übungsaufgabe „üben“

Bevor man die Bildschirmaufnahme startet und in Unity „drauf los“ klickt, sollte man sich sicher sein, dass man auch wirklich genau weiß, wie die Übung umzusetzen ist. In der Bildschirmaufnahme soll schließlich nichts vorgemacht werden, das im Nachhinein ein Fehler war und wieder abgeändert oder gar gelöscht wird. Kleinere Fehler (z.B., wenn man sich verklickt) können zwar im Nachgang noch mit Hilfe eines Videobearbeitungsprogramms herausgeschnitten werden. Jedoch bedeutet das auch immer zusätzliche Arbeit und birgt die Gefahr, dass durch den Videoschnitt kleine Sprünge entstehen, die den Anwender ggf. verwirren könnten.

Daher sollte die Übungsaufgabe vor der Bildschirmaufnahme wenigstens ein Mal durchgeführt und Stolperstellen identifiziert und verinnerlicht werden.

Auf diesem Wege werden somit also auch automatisch Stolperstellen oder sogar Fehler in den bereits bestehenden Unity-Übungsaufgaben auffallen. Auch solche Dinge sollen im Rahmen dieser Arbeit notiert und verbessert bzw. angepasst werden.

## 4. Praktische Ausarbeitung

Dieses Kapitel beinhaltet die Definition der Zielgruppe, die mit dem geplanten E-Learning arbeiten wird, die Analyse des Ist-Zustands der bestehenden Vorlesungsunterlagen, die im Rahmen dieser Arbeit zur Verfügung gestellt wurden und die Definition des Soll-Konzepts für das zu erstellende E-Learning,

### 4.1. Definition der Zielgruppe

Bevor man loslegen kann, muss man sich zunächst überlegen, für welche Zielgruppe das E-Learning überhaupt erstellt werden soll. Welche Personen werden damit arbeiten? Welche technische Ausstattung haben sie? Wie viel Zeit steht ihnen zur Verfügung? Wie ist ihr Wissensstand?

Im Falle dieser Arbeit handelt es sich um Studierende der Hochschule Karlsruhe im Studiengang Kommunikation und Medienmanagement, die die Lehrveranstaltung „Multimedia“ besuchen. Betrachten wir die Eigenschaften dieser Studierenden hinsichtlich des geplanten E-Learnings nun etwas genauer:

- **Alter:** 18-24
- **Geschlecht:** alle Geschlechter
- **Bildungsstand:** Haben grundlegende Programmierkenntnisse, aber ansonsten keine Ahnung von dem Programm
- **Technische Ausstattung:** Poolräume der Hochschule – PCs mit jeweils einem relativ großen Monitor, einer Tastatur und einer Maus. Keine Lautsprecher. Kein Arbeiten von zu Hause aus möglich, da das Programm nicht auf den privaten PCs installiert ist.
- **Zeit:** Während der Vorlesung, in Ausnahmefällen auch kurze Zeit außerhalb der Vorlesung.

### Anforderungen an das E-Learning hinsichtlich der Zielgruppe

Unsere Zielgruppe ist relativ jung, besitzt grundlegende Programmierkenntnisse und kann nur in den Poolräumen während der Vorlesung mit Unity arbeiten.

Was bedeutet das nun für das geplante E-Learning?

Was man daraus ableiten kann, ist, dass die Studierenden sich gut genug auskennen, um ein E-Learning zu benutzen. Man muss ihnen also in der Regel nicht zeigen, wie man das E-Learning pausiert oder zur nächsten bzw. vorherigen Folie springt.

Des Weiteren sollte aufgrund der nur begrenzt zur Verfügung stehenden Zeit das E-Learning so kurzgehalten werden, wie möglich. Allerdings ohne dabei Lücken zu verursachen, die die Studierenden ins Stolpern bringen könnten.

Aufgrund der Tatsache, dass den Studierenden keine Lautsprecher zur Verfügung stehen, sollte für die Vermittlung des Wissens im E-Learning keine Audioausgabe verwendet werden. Stattdessen sollten alle Vorgänge schon aus den Lernvideos heraus nachvollziehbar sein. Eventuelle unklare Situationen können an den entsprechenden Stellen per Textfeld erläutert werden.

## 4.2. Ist-Analyse der Vorlesungsunterlagen

Im Rahmen dieser Arbeit wurden Vorlesungsunterlagen zur Verfügung gestellt, auf denen die geplanten E-Learnings basieren sollen. Bei diesen Vorlesungsunterlagen handelt es sich um die folgenden PDF-Dokumente:

- Unity\_Einstieg.pdf
- Unity\_CS\_Texteingabe\_Textausgabe.pdf
- Unity\_Collision\_Detection.pdf
- Unity\_CS\_ClickCount\_3DObjects.pdf
- Unity\_CS\_Slider\_size\_and\_move.pdf
- Unity\_CS\_Tastensteuerung\_WASD\_qe.pdf
- Unity\_CS\_XML.pdf

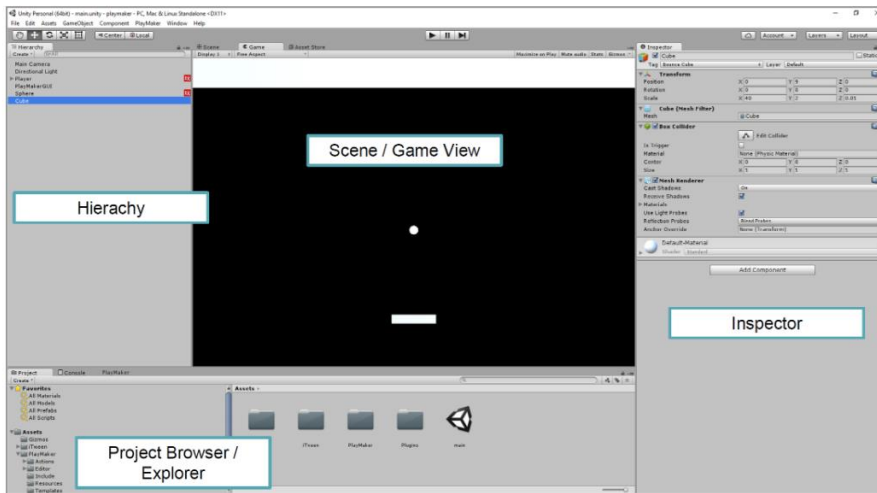
### Unity Einstieg

Die erste Unity-Übung, die ich auf ihre Vollständigkeit, Korrektheit und Durchführbarkeit (im Folgenden Usability genannt) getestet habe, war die Übung aus dem Dokument „Unity\_Einstieg.pdf“. In diesem Dokument wird dem Anwender zunächst kurz geschildert, was genau das Ziel der Übung ist. Es soll ein einfaches 3D-Spiel „mit einem simplen Fahrzeug und einer Fläche, die mit vier Wänden umgeben ist“ erstellt werden (vgl. Schober 2019:02).

### *Aufbau*

Bevor mit der Übungsaufgabe begonnen wird, werden dem Anwender sämtliche Grundfunktionen und Kurzbefehle von Unity erläutert. So wird beispielsweise erklärt, wie man die Kamera ausrichten kann, aus welchen Programmregionen die Systemoberfläche besteht oder über welche Tastenkombinationen im 3D-Raum navigiert werden kann.

## Die Programmregionen



TRM120/130

Prof. Dipl.-Ing. Martin Schober

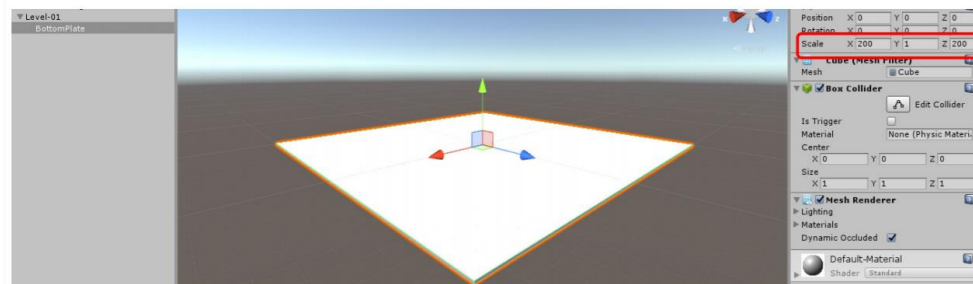
Abbildung 1: Screenshot aus *Unity\_Einstieg.pdf* – Die Programmregionen (Schober 2019:7)

Erst im Anschluss daran, wird tatsächlich mit der ersten Übungsaufgabe begonnen. Hierbei wird vom Anlegen des Projekts bis hin zum fertigen 3D-Spiel jeder Schritt ganz genau erklärt.

## Erstellen der Umgebung für Level-01

Wir erstellen ein leeres Spielobjekt in dem wir die Umgebung erzeugen

1. *GameObject* → *CreateEmpty*
2. Umbenennen des Objekts in Level-01 durch Doppelklick auf den Namen und Überschreiben des Namens
3. *GameObject* → *3D-Object* → *Cube* (Aus dem Cube formen wir eine Grundplatte)
4. *Inspector* → *Transform* → *Scale* (200, 1, 200) = (x, y, z)



TRM120/130

Prof. Dipl.-Ing. Martin Schober

Abbildung 2: Screenshot aus *Unity\_Einstieg.pdf* – Erstellen der Umgebung für Level-01 (Schober 2019:19)

## Usability

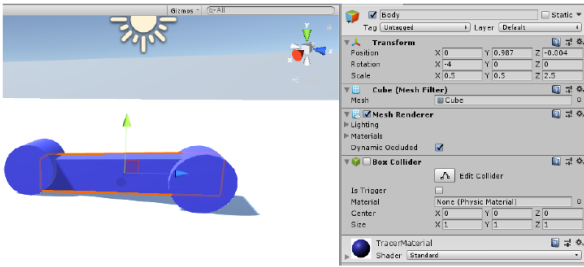
Um zu testen, wie gut sich die Übung umsetzen lässt, ist die beste und effektivste Methode, die Übung direkt praktisch in Unity durchzuführen. Was allerdings vorab schon festgehalten werden kann, ist, dass der Aufbau an sich schon sehr gut durchdacht ist. Durch die Definition des Ziels direkt zu Anfang der Übung, weiß der Anwender direkt, womit er es zutun hat und weiß, was ihn in dieser Übung erwartet.

Die anschließende kurze Erläuterung der Grundfunktionen von Unity trägt außerdem dazu bei, dass der Anwender einen ersten kleinen Überblick über das Programm bekommt und sich schonmal mit der Systemumgebung ein wenig vertraut machen kann.

Die praktische Durchführung der Übung war weitgehend ohne größere Stolperstellen machbar. Die erste Stelle, an der ich hängengeblieben bin, war der 5. Handlungsschritt im Kapitel „Tracer einfärben“.

## Tracer einfärben

1. *Assets* → *Create* → *Material*
2. Einen Materialnamen vergeben z. B. TracerMaterial
3. Das Material im *Inspector* → *Materials* → *Element 0* auf die drei Komponenten verteilen
4. Die Komponenten über der Bodenplatte positionieren, falls nicht schon geschehen
5. Den Body auf *Scale* = (0.5, 1, 3) setzen
6. Die Räder anpassen (Rad in *Hierarchy* aktivieren und *Mesh Renderer* im *Inspector* wieder einschalten)
7. Die Räder in Größe und Position einstellen



TRM120/130
Prof. Dipl.-Ing. Martin Schober

Abbildung 3: Screenshot aus *Unity\_Einstieg.pdf* – Tracer einfärben (Schober 2019:24)

Hier steht geschrieben, dass der Body auf  $Scale = (0.5, 1, 3)$  gesetzt werden soll. Im darauffolgenden Screenshot sieht man den fertig modellierten Tracer und die zugehörigen Werte. Diese Werte und auch das Aussehen des Tracers stimmen nicht mit den Werten überein, die im 5. Handlungsschritt vorgegeben werden. Im Screenshot beträgt der Body-Scale nämlich die Werte  $(0.5, 0.5, 2.5)$ .

Als ich die Übung zunächst mit den Werten, die im 5. Handlungsschritt angegeben werden, durchgeführt habe, fiel mir auf, dass mein Tracer anders aussieht als der Tracer, der im Screenshot zu sehen ist.

Das Problem hierbei ist, dass Unstimmigkeiten dieser Art den Anwender immer sehr schnell innehalten lassen und können viel Zeit kosten, da der Anwender ggf. versucht, herauszufinden, ob er einen Fehler gemacht hat und macht aufgrund dessen ggf. nicht weiter, bis sein Zwischenergebnis dem aus der Anleitung entspricht. Wenn man Screenshots als Zwischenergebnis einbindet, so sollte immer genauestens darauf geachtet werden, dass hier genau das angezeigt wird, was auch der Anwender vor Augen haben wird. Falls eine Übung aufgrund von beispielsweise Systemupdates angepasst werden muss, so muss auch darauf geachtet werden, dass sämtliche Screenshots überprüft und ggf. angepasst bzw. entfernt werden. Gar keine Grafik zu verwenden ist besser, als eine zu verwenden, in der veraltete Dinge abgebildet sind, die den Anwender in die Irre leiten könnten.

Das gleiche Problem ereignet sich ein paar Handlungsschritte später erneut. Es soll ein Box Collider für das Gefährt erstellt werden.



Abbildung 4: Screenshot aus *Unity\_Einstieg.pdf* – Einen Box Collider für das Gefährt erstellen (Schober 2019:27)

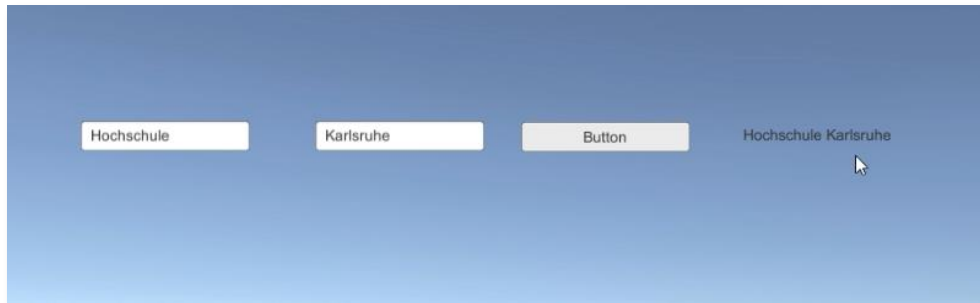
Im 3. Handlungsschritt soll die Größe des Box Colliders mit den Werten (2, 1, 3.8) definiert werden. Im darauffolgenden Screenshot sieht man allerdings, dass für die Größe die Werte (0.7, 0.7, 2.8) verwendet wurden. Hier werden also wieder unterschiedliche Dinge angegeben.

Außerdem fällt auf, dass der Box Collider mit den Werten aus dem Screenshot eigentlich etwas zu klein für das Fahrzeug ist. Wenn das Fahrzeug so wie es im Screenshot abgebildet ist, beispielsweise mit dem äußeren Bereich des Vorderrads gegen ein Hindernis fahren würde, so würde keine Kollision erfolgen. Der Teil des Rads, der sich außerhalb des Box Colliders befindet, ist widerstandslos.

Für das geplante E-Learning sollten diese Werte also am besten ein wenig angepasst werden, sodass der Box Collider der Größe des Fahrzeugs entspricht.

## Unity + C#: Texte eingeben/ausgeben durch Mausklick

In dieser Übung sollen zwei Textfelder erstellt werden, in die man beispielsweise jeweils einen Vor- und einen Nachnamen eingeben kann. Beim Klick auf einen Button soll anschließend der Inhalt der Textfelder kombiniert ausgegeben werden.



*Abbildung 5: Ergebnis der Übung „Texte eingeben/ausgeben durch Mausklick“*

### Aufbau

Das Dokument mit der zweiten Übung (Unity\_CS\_Texteingabe\_Textausgabe.pdf) ist verglichen mit dem Dokument der ersten Übung deutlich kürzer gehalten. Logischerweise wird hier davon ausgegangen, dass im Voraus bereits das Dokument „Unity Einstieg“ gelesen und die Übung durchgeführt wurde. Dementsprechend müssen dem Anwender auch nicht erneut die Grundfunktionen und die Toolumgebung von Unity erklärt werden. Wenn der Anwender vergessen sollte, wie etwas aus der ersten Übung funktioniert hat oder welches das richtige Tastenkürzel ist, dann kann er das bei Bedarf nochmals gezielt im entsprechenden Dokument nachschauen.

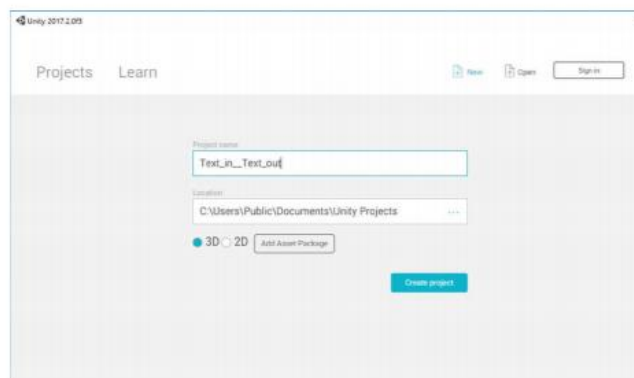
Es wird also ohne Umschweife direkt mit der Übungsaufgabe begonnen. Was hierbei negativ auffällt, ist, dass auch auf die Definition des Ziels der Übung verzichtet wurde. Der Anwender weiß also nicht genau, was ihn in dieser Übung erwarten wird. Er kann lediglich aus der Überschrift ableiten, dass es sich wohl um irgendeine Art Text-Ein- und Text-Ausgabe handeln muss.

## Unity + C#: Texte eingeben/ausgeben durch Mausklick

Prof. Dipl.-Ing. Martin Schober

### Unity + C#: Texte eingeben/ausgeben durch Mausklick

1. Starten Sie Unity
2. Vergeben Sie einen Projektnamen



Sloan Kelly Beispiele!

Multimedia Engineering

Prof. Dipl.-Ing. Martin Schober

**Abbildung 6:** Screenshot aus *Unity\_CS\_Texteingabe\_Textausgabe.pdf* – Texte eingeben/ausgeben durch Mausklick (Schober 2019:1)

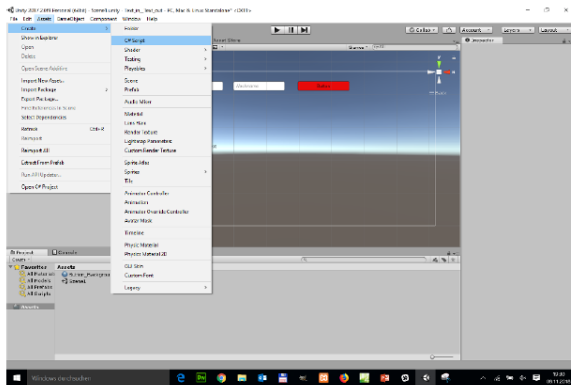
#### *Usability*

Aufgrund der Tatsache, dass die Zieldefinition in dieser Übung fehlt, steht man als Anwender dieser Übung ahnungslos und skeptisch gegenüber. Der Anwender weiß vorab nicht, was der Sinn der Übung ist und was er nach der Durchführung Neues gelernt hat. Es fehlt also komplett der Kontext.

Die praktische Durchführung dieser Übung war auch hier wieder zunächst ohne größere Stolperstellen machbar. Die erste Stelle, an der ich hängengeblieben bin, war das Einbinden des C#-Scripts.

Zunächst soll der Anwender ein C#-Script auf die gleiche Art und Weise anlegen, wie zuvor bereits in der ersten Übung. (siehe Abbildung 7)

## C#-Skript anlegen



Menue → Assets → Create → C#-Skript

Multimedia Engineering

Prof. Dipl.-Ing. Martin Schober

Abbildung 7: Screenshot aus *Unity\_CS\_Texteingabe\_Textausgabe.pdf* – C#-Skript anlegen (Schober 2019:8)

Diesmal wurde allerdings nicht dazugeschrieben, wie das Skript benannt werden soll. Das kann zu Problemen führen, wenn das Skript über die Benennung von einem anderen Skript heraus aufgerufen werden sollte.

Wer genau aufpasst, kann beim kurz darauffolgenden Screenshot vom C#-Skript (siehe Abbildung 8) erkennen, dass das Skript „Ausgeben.cs“ heißt. Man sollte aber nicht davon ausgehen, dass die Studierenden auf diese Kleinigkeit achten werden.

## Das Skript schreiben...

```

Ausgeben.cs
Ausgeben { Absenden ()
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI; // Dieser Eintrag muss für die Nutzung von UI-Komponenten ergänzt werden!
5
6 public class Ausgeben : MonoBehaviour { // Klassennamen übernehmen oder neu vergeben.
7
8     public InputField name1; // Variable die Zeichen aus Textfeld 1 übernimmt.
9     public InputField name2; // Variable die Zeichen aus Textfeld 2 übernimmt.
10    public Text all_names; // Variable die Zeichen aus Textfeld 1 und Textfeld 2 übernimmt.
11
12    public void Absenden() // Eine Funktion anlegen, die die Texte zusammenfasst in eine Variable.
13        // und die Ausgabe in der Console und in das dritte Textfeld steuert.
14    {
15        Debug.Log (name1.text); // Ausgabe in Console
16
17        all_names.text = name1.text + " " + name2.text; // Ausgabe in das dritte Textfeld.
18
19    }
20
21 }
22
23
    
```

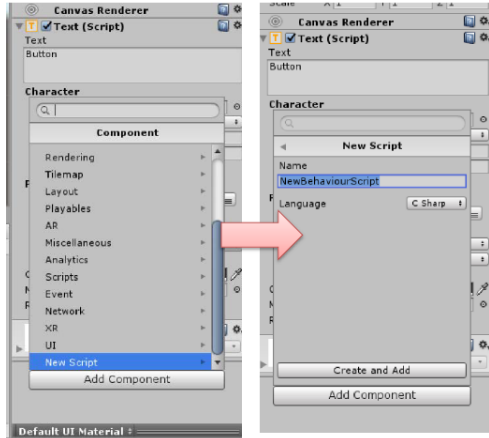
Multimedia Engineering

Prof. Dipl.-Ing. Martin Schober

Abbildung 8: Screenshot aus *Unity\_CS\_Texteingabe\_Textausgabe.pdf* – Das Skript schreiben (Schober 2019:10)

Direkt im Anschluss daran soll ein weiteres C#-Skript hinzugefügt werden. Diesmal allerdings über einen anderen Weg.

## Dem Button ein C#-Skript hinzufügen



- Im Eigenschaftsfenster am rechten Rand auf Add Component klicken!
- New Script auswählen!
- Einen Namen vergeben.

Multimedia Engineering

Prof. Dipl.-Ing. Martin Schober

**Abbildung 9:** Screenshot aus *Unity\_CS\_Texteingabe\_Textausgabe.pdf* – Dem Button ein C#-Skript hinzufügen (Schober 2019:9)

Beim Durchführen dieses Handlungsschritts bin ich sehr lange stecken geblieben. Das lag daran, dass nicht ersichtlich war, wofür genau dieses Skript sein soll und welchen Inhalt das Skript haben soll. Der Quelltext aus dem Screenshot in Abbildung 8 könnte sowohl für das normal angelegte Skript aus Abbildung 7 als auch für das speziell angelegte Skript aus Abbildung 9 sein. Da ich nicht wusste, in welches der beiden Skripte der Quellcode nun hineingeschrieben werden muss, habe ich zunächst einfach beide Skripte mit diesem Quellcode befüllt. Dies ergab jedoch eine Fehlermeldung in der Konsole, weshalb ich eine ganze Weile herumprobiert habe, wie sich das Problem lösen lässt. Nach einiger Zeit stellte ich fest, dass alles einwandfrei funktionierte, sobald ich das Skript aus Abbildung 9 entfernt habe.

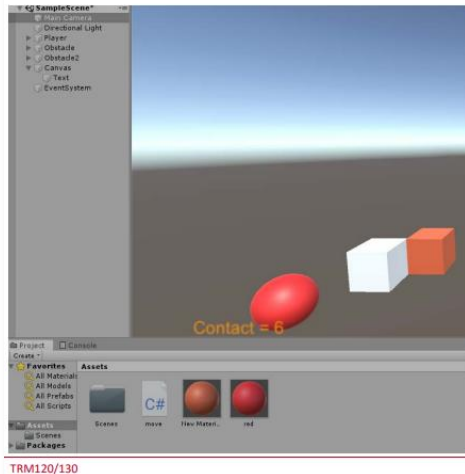
## Unity Collision Detection

In dieser Übung sollen ein weißer Würfel, ein orangener Würfel und eine rote Kugel erstellt werden. Der weiße Würfel soll über die Pfeiltasten gesteuert werden können. Sämtliche Berührungen des weißen Würfels mit den anderen beiden Objekten sollen gezählt und in einem Textfeld ausgegeben werden.

### Aufbau

Das Dokument mit der dritten Übung (*Unity\_Collision\_Detection.pdf*) ist genau wie die zweite Übung deutlich kürzer gehalten. Auch hier wird wieder davon ausgegangen, dass im Voraus bereits das Dokument „Unity Einstieg“ gelesen und die Übung durchgeführt wurde. Auch in diesem Dokument fehlt wieder die Zielsetzung. Stattdessen gibt es vor Beginn der Übung eine Folie, in der der Aufbau der Szene geschildert wird.

## Aufbau der Szene



1. Der weiße Würfel kann über die Pfeiltasten gesteuert werden.
2. Der orangefarbene Würfel und die rote Kugel sind Objekte die vom weißen Würfel berührt werden können.
3. Die Berührungen sollen erkannt und gezählt werden.
4. Die Anzahl der Berührungen sollen in einem Textfeld ausgegeben werden.

Abbildung 10: Screenshot aus *Unity\_Collision\_Detection.pdf* – Aufbau der Szene (Schober 2019:3)

Hier wird erklärt, was gemacht werden soll. Somit startet der Anwender zwar nicht komplett ahnungslos in die Übung (wie in Übung 2), allerdings weiß er auch hier nicht wirklich, was genau der Sinn hinter dieser Übung sein soll. Der Kontext fehlt also auch hier.

### Usability

Die Übung beginnt bereits bei der Folie „Aufbau der Szene“. Hier wird also nicht nur erläutert, was in dieser Übung zu tun ist, sondern es wird auch direkt davon ausgegangen, dass der Anwender hier bereits beginnt, die geforderten Komponenten (zwei Würfel und eine Kugel) und das zugehörige Material (orange und rot) zu erstellen. Es gibt keine Folie, in der erklärt wird, wie man diese Komponenten und das Material erstellt. Es wird also davon ausgegangen, dass man noch aus den vorherigen Übungen weiß, wie das funktioniert und dieses Wissen wird für diese Übungsaufgabe vorausgesetzt.

Für mich hat diese Vorgehensweise kein Problem dargestellt. Es könnte aber durchaus vorkommen, dass andere Anwender hier nicht wissen, was sie tun sollen. Zum Beispiel wäre das der Fall, wenn sie die vorherigen Übungen aus irgendeinem Grund übersprungen haben sollten.

Daher wäre es an dieser Stelle angebracht, einen Hinweis zu ergänzen, in dem steht, in welchem Dokument und auf welcher Seite man die Erstellung von 3D-Objekten und das Anlegen und Zuweisen eines Materials nachlesen kann. In diesem Fall befinden sich diese Informationen im Dokument „Unity\_Einstieg.pdf“ auf Seite 22 bzw. S. 24.

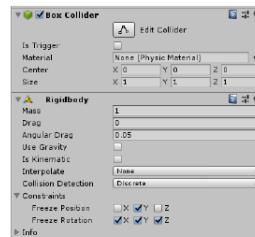
Nachdem man als Anwender die geforderten Objekte erstellt und mit dem entsprechenden Material eingefärbt hat, folgt eine kurze Info-Folie. Auf dieser Folie wird erklärt, wie die Collision Detection funktioniert und was hierfür die Voraussetzungen sind (siehe Abbildung 11).

## Collision Detection

In der Unity Physics Engine integriert ist die „Collision Detection“.

Erst wenn man dem GameObject eine **Collider**-Komponente hinzufügt, werden Berührungen mit anderen Collidern registriert. Die Physics Engine ist wie eine unsichtbare Welt im Hintergrund, in der nur Objekte vorkommen, die mit entsprechenden Komponenten ausgerüstet sind.

GameObjects mit einem Collider, die nicht nur an einer Position von Anfang bis zum Ende verharren, sondern sich auch in Bewegung setzen, müssen zusätzlich eine **Rigidbody**-Komponente haben. Diese dient Unity bei der Optimierung der Performance von Collidern und ist Voraussetzung um eine Kollision aus der Bewegung heraus auszulösen. Wenn zwei Objekt aufeinander treffen muss also mindestens eines der beiden einen Rigidbody haben, damit die **Events** im Code ausgelöst werden.



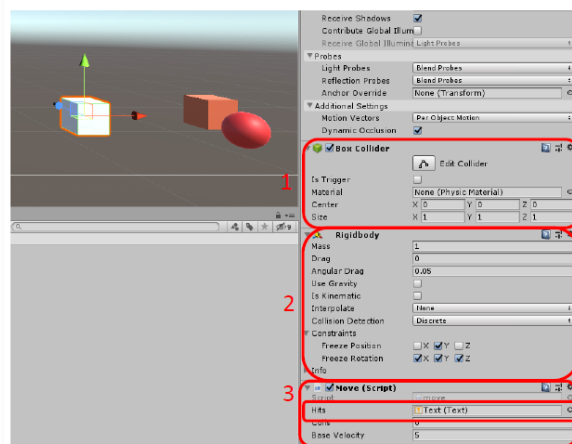
TRM120/130

Prof. Dipl.-Ing. Martin Schober

Abbildung 11: Screenshot aus Unity\_Collision\_Detection.pdf – Collision Detection (Schober 2019:4)

Die darauffolgende Folie beinhaltet eine Handlungsanweisung, die aus vier Handlungsschritten besteht. Diese Handlungsanweisung basiert auf der Tatsache, dass der Anwender bereits einen Rigidbody für den weißen Würfel angelegt hat. Dass ein Rigidbody für den weißen Würfel angelegt werden muss, wird zuvor nirgends erwähnt. Man muss als Anwender selbst darauf kommen, indem man sich den Screenshot (siehe Abbildung 12) genau anschaut und sieht, dass der weiße Würfel ausgewählt ist und im Inspector Rigidbody-Eigenschaften zu sehen sind.

## Rigidbody



Einstellungen Rigidbody des weißen Cube:

1. Box Collider muss aktiv sein
2. Use Gravity deaktivieren und Constraints wie im Bild aktivieren
3. Nach dem Erstellen des C#-Skripts wird das Script aus dem Assets-Bereich auf den aktivierten weißen Cube gezogen. Daraufhin erscheint das Script im Inspector des Objekts.
4. Sobald das Script und das Textfeld erstellt sind, wird der entsprechenden Variablen, hier „Hits“ das Textfeld zugewiesen. Die Zuweisung erfolgt mit der Auswahl, die erscheint, wenn der kleine Kreis rechts des Textfeldes geklickt wird.

TRM120/130

Prof. Dipl.-Ing. Martin Schober

Abbildung 12: Screenshot aus Unity\_Collision\_Detection.pdf – Rigidbody (Schober 2019:5)

Diese Eigenschaften erscheinen erst, wenn man für das Objekt einen Rigidbody angelegt hat.

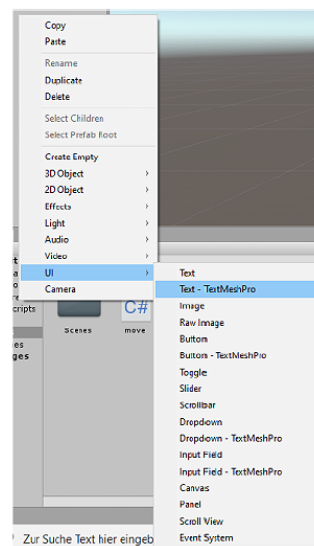
Auch hier könnten wieder einige Anwender zunächst hängenbleiben. Daher wäre es auch an dieser Stelle angebracht, einen Hinweis zu ergänzen, in dem steht, in welchem Dokument und auf welcher Seite man das Anlegen eines Rigidbodys finden kann. In diesem Fall befinden sich diese Informationen im Dokument „Unity\_Einstieg.pdf“ auf Seite 28.

Im 3. Handlungsschritt wird erneut etwas vorweggenommen. Der Handlungsschritt beginnt mit den Worten: „Nach dem Erstellen des C#-Skripts“. Allerdings hat man als Anwender zu diesem Zeitpunkt noch kein C#-Skript angelegt und muss somit aus der Handlungsanweisung herauspringen und schauen, ob das Anlegen des Skripts in den darauffolgenden Folien erläutert wird.

Wenn man nun eine Folie weiter geht, so wird allerdings nicht erst das Anlegen eines C#-Skripts erläutert, sondern stattdessen folgt hier eine Folie, in der gezeigt wird, wie man ein Textfeld erzeugt. Dieses Textfeld wird im 4. Handlungsschritt aus Abbildung 12 vorausgesetzt.

## Ein Textfeld erzeugen

1. Mit der rechten Maustaste  
In das **create**-Fenster klicken und  
anschließend **UI** und **TextMeshPro**  
auswählen



TRM120/130

Prof. Dipl.-Ing. Martin Schober

Abbildung 13: Screenshot aus Unity\_Collision\_Detection.pdf – Ein Textfeld erzeugen (Schober 2019:6)

Rein chronologisch betrachtet hätte allerdings zunächst das C#-Skript und danach erst das Anlegen eines Textfeldes erläutert werden müssen. Diese Folien sollten also am besten getauscht werden.

Beim Durchführen der Übung bin ich an dieser Stelle dann auch längere Zeit hängengeblieben, da in der Handlungsanweisung geschrieben steht, dass man ein „TextMeshPro“-Textfeld erzeugen soll. Mit dieser Art von Textfeld ließ sich die Übung allerdings nicht umsetzen. Erst als ich ein normales Textfeld anstatt dem TextMeshPro-Textfeld verwendet habe, funktionierte alles so, wie es soll.

## Zusammenfassung

### Unity Einstieg

(Unity\_Einstieg.pdf)

#### Probleme

- ⊗ Werte im Handlungsschritt und Werte aus dem Screenshot stimmen nicht überein.
- ⊗ Dadurch stimmt das Zwischenergebnis, das im Screenshot abgebildet ist, nicht mit dem Zwischenergebnis des Anwenders überein.
- ⊗ Die angegebenen Werte für den Box Collider sind zu niedrig. Dadurch ist der Box Collider eigentlich zu klein für das Fahrzeug.

#### Maßnahmen

- ✔ Screenshots bei Änderungen immer auf Aktualität prüfen und ggf. neu erstellen.
- ✔ Kein Screenshot ist besser als ein falscher bzw. nicht mehr aktueller Screenshot
- ✔ Angegebene Werte immer gegenprüfen.

### Unity + C#: Texte eingeben/ausgeben durch Mausklick

(Unity\_CS\_Texteingabe\_Textausgabe.pdf)

#### Probleme

- ⊗ Es gibt keine Zieldefinition für die Übung. Der Kontext fehlt komplett.
- ⊗ Es gibt keine Vorgabe, wie das C#-Skript benannt werden soll. Dies kann zu Problemen führen.
- ⊗ Es wird ein weiteres C#-Skript angelegt – auf eine andere Art und Weise als das erste Skript und ohne Erklärung, wofür dieses 2. Skript sein soll.
- ⊗ Aus der Folie, die den Quelltext für eines der Skripte beinhaltet, wird nicht ersichtlich, für welches der beiden Skripte der Quelltext verwendet werden soll.
- ⊗ Das im Nachhinein angelegte 2. Skript hat keinen ersichtlichen Nutzen für die Übung. Erst wenn das Skript komplett entfernt wird, funktioniert alles so, wie gewünscht.

### Maßnahmen

- ✔ Zieldefinition ergänzen
- ✔ Keine wichtigen Handlungsschritte weglassen. In diesem Fall ergänzen, wie das C#-Skript benannt werden muss.
- ✔ Bei neuen, ungewöhnlichen Handlungsschritten erklären, was der Sinn dahinter ist.
- ✔ Wenn es mehrere Skripte gibt, dann muss für den vorgegebenen Quelltext festgelegt werden, für welches Skript dieser gültig ist.
- ✔ Den Handlungsschritt bezüglich des Anlegens eines 2. Skripts komplett entfernen, da die Übung erst ohne das hier angelegte Skript fehlerfrei funktioniert.

## Unity + C#: Klicks auf 3D-Objekte zählen und ausgeben

(*Unity\_Collision\_Detection.pdf*)

### Probleme

- ✘ Es gibt keine Zieldefinition für die Übung. Der Kontext fehlt komplett.
- ✘ Es werden Handlungsschritte vorausgesetzt, die nirgends erläutert werden.
- ✘ Es gibt keinen Hinweis, wo bzw. in welchem Dokument man für die Aufgabe notwendige Handlungsschritte nochmal nachlesen kann.
- ✘ Es gibt Handlungsanweisungen in nicht chronologischer Reihenfolge.

### Maßnahmen

- ✔ Zieldefinition ergänzen
- ✔ Für die Übung notwendige Handlungsschritte entweder ergänzen oder einen Hinweis hinzufügen, wo diese nachgelesen werden können.
- ✔ Handlungsanweisungen immer in chronologischer Reihenfolge anordnen.

## 4.3. Soll-Konzept für das geplante E-Learning

Nachdem die vorhandenen Übungsaufgaben aus den Vorlesungsunterlagen durchgeführt und hinsichtlich der Usability getestet wurden, folgt die Ausarbeitung eines Soll-Konzepts für das geplante E-Learning.

In diesem Soll-Konzept werden alle in Kapitel 4.2 festgestellten Fehler und Probleme in der Theorie korrigiert und verbessert. Die Maßnahmen, um die in den Vorlesungsunterlagen gefundenen Probleme zu beheben, wurden im vorherigen Kapitel bereits aufgezeigt. Jetzt sollen zusätzlich die geplanten Technologien zur Umsetzung des E-Learnings beachtet und in die Konzeptdefinition miteinbezogen werden.

## Geplanter Aufbau

### Unity Einstieg

Das Dokument Unity Einstieg ist insgesamt ein sehr langes Dokument. Hier werden zunächst die grundlegenden Funktionen und die Systemoberfläche von Unity erklärt und darauf folgt eine ziemlich umfangreiche erste Übung.

Im Kapitel 3.1 Wissen vermitteln wurde bereits über die verschiedenen Möglichkeiten der Motivationssteigerung gesprochen. Hier wurde unter anderem festgelegt, dass zu lange Lerneinheiten auf den Anwender demotivierend wirken und eine lange Lerneinheit daher lieber in 2 oder mehrere kürzere Lerneinheiten aufgeteilt werden sollten.

Daher bietet es sich bei der Übung Unity Einstieg an, diese in 2 E-Learnings aufzuteilen. Im ersten E-Learning können dann alle grundlegenden Funktionen und die verschiedenen Tastenkürzel gezeigt werden und das zweite E-Learning beinhaltet dann die Übungsaufgabe.

Diese Aufteilung bringt zum einen den Vorteil, dass der Anwender aufgrund der insgesamt kürzeren E-Learnings motivierter bei der Sache ist, und zum anderen, dass er auch bei allen später folgenden E-Learnings jederzeit weiß, dass er grundlegende Funktionen im ersten E-Learning finden und nochmals nachsehen kann.

#### *Geplante Gliederung für das erste E-Learning „Unity Einstieg“:*

1. Start-Folie, die den Titel des E-Learnings und die Unity-Version enthält
2. Anleitende Folie: die **Kamera ausrichten**
3. Beschreibende Folie, in der die **Systemoberfläche** erklärt wird
4. Beschreibende Folie, in der die **Programmregionen** gezeigt werden
5. Beschreibende Folie, in der die **Hierarchy**-Programmregion genauer erklärt wird
6. Beschreibende Folie, in der die **Scene** mit ihrer **Toolbar** und das **Gizmo** erläutert werden
7. Beschreibende Folie, in der die **Toolbar** genauer erklärt wird
8. Beschreibende Folie, in der die **Toolbar** mit den grundlegenden Bedienelementen genauer erklärt wird
9. Beschreibende Folie, in der die **Scene-Toolbar** erklärt wird

10. Beschreibende Folie, in der die **Inspector**-Programmregion genauer erklärt wird
11. Beschreibende Folie, in der die **Project-Browser**-Programmregion genauer erklärt wird
12. Beschreibende Folie, in der die **Project-Browser-Console** genauer erklärt wird
13. Anleitende Folie: im 3D-Raum um das **aktuelle Zentrum rotieren**
14. Anleitende Folie: im 3D-Raum die **Kamera bewegen**
15. Anleitende Folie: im 3D-Raum **zoomen**
16. Anleitende Folie: im 3D-Raum mit den **Pfeiltasten navigieren**
17. Anleitende Folie: im 3D-Raum den **Flugmodus** bedienen
18. Anleitende Folie: im 3D-Raum **Objekte** automatisch in **den Fokus** nehmen

*Geplante Gliederung für das zweite E-Learning „Unity Einstieg – Übung“:*

1. Start-Folie, die den Titel des E-Learnings und die Unity-Version enthält
2. Info-Folie, die die Zielsetzung des E-Learnings enthält
3. Anleitende Folie: ein neues Projekt anlegen
4. Anleitende Folie: eine Umgebung für Level-01 erstellen
5. Anleitende Folie: eine **Grundplatte** erstellen
6. Anleitende Folie: die **Nordwand** bauen
7. Anleitende Folie: die **Südwand** bauen
8. Anleitende Folie: die **Ostwand** bauen
9. Anleitende Folie: die **Westwand** bauen
10. Anleitende Folie: die **Objekte für das Fahrzeug** erstellen
11. Anleitende Folie: die **Collider** entfernen
12. Anleitende Folie: ein **Material** erstellen und den **Tracer** damit **einfärben**
13. Anleitende Folie: die **Komponenten über der Bodenplatte** positionieren
14. Anleitende Folie: den **Body** und die **Räder modellieren**
15. Anleitende Folie: einen **Box Collider** für das Fahrzeug erstellen
16. Anleitende Folie: das Fahrzeug mit einem **C#-Programm** in Bewegung setzen
17. Anleitende Folie: das **Spiel mit Build als Anwendung** erstellen
18. **Troubleshooting-Folie**, in der gezeigt wird, was man tun muss, falls das Fahrzeug falschherum fährt

Bei der letzten Folie handelt es sich um ein Troubleshooting, das zeigt, was man tun muss, falls das Fahrzeug falschherum fahren sollte. Dieses Troubleshooting existiert in der ursprünglichen Übungsaufgabe nicht. Als ich die Übung zum ersten Mal durchgeführt habe, ist bei mir aber genau das passiert – das Auto fuhr falschherum, beziehungsweise rückwärts. Da dieses Problem mit hoher Wahrscheinlichkeit auch bei einigen Studierenden auftreten könnte, die diese Übung durchführen, habe ich mich dazu entschieden, ein Troubleshooting zu ergänzen. Hier wird gezeigt, wie man in nur wenigen Schritten das Problem mit dem Rückwärtsfahren beheben kann.

## Unity + C#: Texte eingeben/ausgeben durch Mausklick

Das Dokument mit der zweiten Übung ist im Vergleich zur ersten Übung nicht so lang, dass sie in zwei E-Learnings aufgeteilt werden müsste. Hier kann also direkt mit der Übung begonnen werden.

### *Geplante Gliederung für das erste E-Learning „Unity + C#: Texte eingeben/ausgeben durch Mausklick“:*

1. Start-Folie, die den Titel des E-Learnings und die Unity-Version enthält
2. Info-Folie, die die Zielsetzung des E-Learnings enthält
3. Anleitende Folie: ein neues Projekt anlegen
4. Anleitende Folie: ein InputField erzeugen
5. Anleitende Folie: die Kamera und das InputField ausrichten
6. Anleitende Folie: das Input-Field duplizieren
7. Anleitende Folie: einen Button erzeugen
8. Anleitende Folie: ein Textfeld für die Ausgabe erstellen
9. Anleitende Folie: ein C#-Skript anlegen
10. Anleitende Folie: C#-Skript in den Button einbinden
11. End-Folie, in der das Ergebnis der Übung gezeigt wird.

## Geplante Technologien zur Umsetzung

Geplant ist, dass das E-Learning in erster Linie Lernvideos (bzw. Tutorial-Videos) beinhalten soll. Bei diesen Lernvideos wird es sich um Screencasts bzw. Bildschirmaufnahmen handeln, die aufgenommen werden, während die Übungsaufgaben von mir in Unity durchgeführt werden.

Um die Bildschirmaufnahmen zu erstellen, gibt es 2 Möglichkeiten, die für mich im Rahmen dieser Arbeit Betracht kommen. Die Bildschirmaufnahme über **GeForce Experience** und die Bildschirmaufnahme über **Adobe Captivate**.

Ich werde beide Möglichkeiten testen und mich dann für eine Variante entscheiden, mit der alle darauffolgenden Bildschirmaufnahmen erstellt werden.

Die fertigen Bildschirmaufnahmen sollen anschließend mit **Adobe Premiere Pro** nachbearbeitet werden. Hierbei sollen beispielsweise längere Passagen, in denen die Maus sich nicht bewegt, herausgeschnitten werden (z.B., wenn man während der Aufnahme kurz stoppt, um nachzusehen, wie man weitermachen muss.)

Des Weiteren soll am Ende jeder Bildschirmaufnahme ein Übergang zu schwarz eingefügt werden. Dadurch enden die Bildschirmaufnahmen nicht abrupt, sondern es entsteht ein angenehmer Übergang zur nächsten Folie.

## 4.4. Umsetzung des Soll-Konzepts

### Erstellen von Lernvideos per Screencast

Schriftliche Anleitungen für komplexe Sachverhalte können schnell sehr unübersichtlich und verwirrend werden. Screencasts bzw. Bildschirmaufnahmen bieten die Möglichkeit, komplexe Prozesse einfach und nachvollziehbar darzustellen.

#### Was ist ein Screencast?

*„Der Begriff Screencast setzt sich aus den englischen Worten „screen“ und „cast“ zusammen und bedeutet übersetzt so viel wie „Bildschirmübertragung“. Ein Screencast ist also eine Videoaufzeichnung Ihres Bildschirms.“ Ionos (2020): „Screencast im Überblick: So funktioniert die Bildschirmaufzeichnung“.*

*<<https://www.ionos.de/digitalguide/online-marketing/verkaufen-im-internet/screencast/>> [Stand: 02.09.2018]*

Bei einem Screencast werden alle Inhalte aufgezeichnet, die auf dem Bildschirm zu sehen sind. Dabei handelt es sich in der Regel um aktive Programme, geöffnete Dokumente und die Mauszeiger-Bewegung.

(vgl. Ionos (2020): „Screencast im Überblick: So funktioniert die Bildschirmaufzeichnung“.

*<<https://www.ionos.de/digitalguide/online-marketing/verkaufen-im-internet/screencast/>> [Stand: 02.09.2018])*

Solche Screencasts eignen sich ideal für Lernvideos, wie sie in dieser Arbeit angestrebt werden.

#### Worauf muss man achten?

Zunächst muss man sich für eine Screencast-Software entscheiden und diese auf seinem Computer installieren. Sobald das erledigt ist, kann theoretisch mit der Bildschirmaufnahme begonnen werden. Dennoch sollte man sich zunächst auf die Bildschirmaufnahme vorbereiten und einen gewissen Plan vor Augen haben.

### *Recherche und Ablauf planen*

Vorab müssen alle notwendigen Informationen für den Screencast recherchiert werden. Im Fall dieser Arbeit wird es sich um einen Screencast handeln, in dem die Systemoberfläche von Unity aufgenommen wird, während Übungsaufgaben durchgeführt werden. Daher sollte die Übungsaufgabe mindestens 1x durchgeführt werden, bevor die Bildschirmaufnahme gestartet wird. Wenn man die Übungsaufgabe nicht vorher übt, dann wird man sehr wahrscheinlich an vielen Stellen hängen bleiben. Das führt zu einer unschönen Aufnahme und muss gegebenenfalls im Nachhinein mit viel Aufwand in einem Videobearbeitungsprogramm wie Adobe Premiere Pro korrigiert werden.

### *Arbeitsplatz einrichten*

Vor der Bildschirmaufnahme sollte sichergestellt werden, dass in der Bildschirmaufnahme nur die Inhalte sichtbar sind, die für den Anwender relevant sind. Das bedeutet, dass vorab der Desktop aufgeräumt werden sollte. Während der Nutzung von Unity kann es nämlich passieren, dass beim Laden für kurze Zeit der Desktop zu sehen ist. Das ist z.B. der Fall, man ein Projekt veröffentlicht. Im Idealfall sollte also ein neutrales Hintergrundbild ausgewählt werden und keine privaten Dateien wie Bilder, Ordner, Spiele etc. auf dem Desktop liegen.



*Abbildung 14: Unaufgeräumter Desktop*



*Abbildung 15: Aufgeräumter Desktop*

### *Screencast erstellen*

Nun kann die Screencast-Software gestartet und alle notwendigen Einstellungen festgelegt werden. Hier sollten unter anderem der Bildausschnitt, diverse Grafikeinstellungen und die Audioeinstellungen festgelegt werden. In diesem Fall kann der Ton für die Bildschirmaufnahmen komplett deaktiviert werden, da die PCs bzw. Monitore in den Poolräumen keine Lautsprecher besitzen.

Sobald alles eingestellt ist, kann die Bildschirmaufnahme gestartet werden.

### *Screencast bearbeiten*

Nachdem der Bildschirm aufgezeichnet wurde, kann man die Aufzeichnung bearbeiten. Je nach Programm kann man die Aufnahme direkt in der Software bearbeiten. Es kann aber auch sein, dass man eine Screencast-Software hat, mit der sich die Aufnahmen im Nachhinein nicht bearbeiten lassen. Wenn das der Fall sein sollte, dann muss für eine anschließende Bearbeitung der Aufnahmen eine separate Videobearbeitungs-Software wie beispielsweise Adobe Premiere Pro genutzt werden.

### *Screencast exportieren*

Im letzten Schritt muss die finale Videodatei nur noch gerendert werden. Hierbei kann in der Regel zwischen verschiedenen Video-Formaten ausgewählt werden. Für diese Arbeit sollen alle Video-Dateien im MP4-Format exportiert werden.

## Screencast mit Adobe Captivate



**Abbildung 16:** Screenshot aus Adobe Captivate 2019: Ein neues Projekt erstellen

Wer mit Adobe Captivate ein E-Learning erstellen und Bildschirmaufnahmen verwenden möchte, der muss sich zusätzlich zu Captivate kein weiteres Screencast-Programm zulegen. Adobe Captivate besitzt nämlich ein bereits implementiertes Tool, mit dem der Bildschirm aufgenommen werden kann. Dieses Tool nennt sich „Videodemo“.

„Videodemos in Adobe Captivate sind eine Möglichkeit, Produktfunktionen im Videoformat zu demonstrieren. Sie können sie auch verwenden, um eine neue Produktfunktion zu demonstrieren.“ Adobe (2018): „Erstellen von Videodemos“. <<https://helpx.adobe.com/de/captivate/how-to/create-video-demo.html>>

[Stand: 23. August 2018]

Wenn man in Adobe Captivate „Videodemo“ auswählt und anschließend auf „Erstellen“ klickt (siehe Abbildung 16), dann öffnet sich das folgende Fenster:

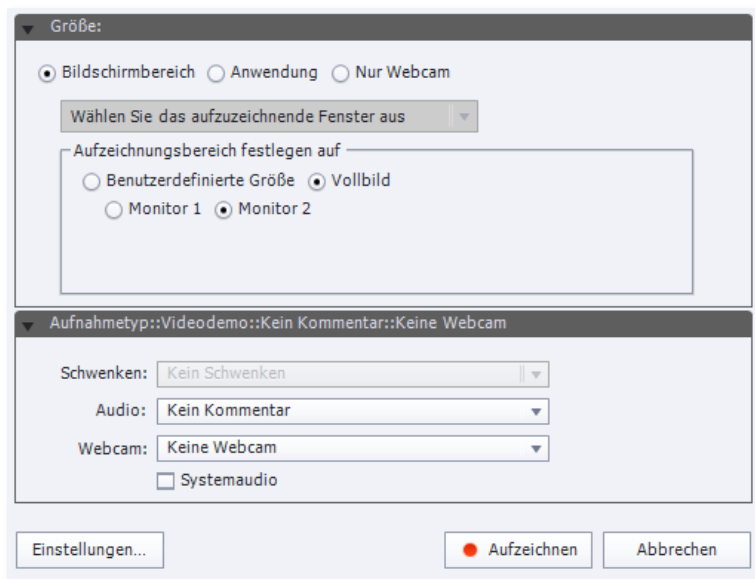


Abbildung 17: Screenshot aus Adobe Captivate 2019: Videodemo Voreinstellungen „Bildschirmbereich“

Wie in Abbildung 17 zu sehen ist, kann man bei den Voreinstellungen für die Videodemo vorab einige Dinge einstellen. Was in unserem Fall besonders wichtig ist, ist die richtige Quelle für die Bildschirmaufnahme auszuwählen. In den Voreinstellungen kann man zwischen „Bildschirmbereich“, „Anwendung“ und „Nur Webcam“ wählen. Letzteres spielt für uns keine Rolle, da wir nicht mit einer Webcam arbeiten.

### **Bildschirmbereich**

Wählt man als Quelle für die Aufzeichnung „Bildschirmbereich“ aus, so hat man zwei Möglichkeiten. Eine „Vollbildaufnahme“, bei der der gesamte Bildschirm aufgenommen wird oder eine „Benutzerdefinierte Größe“, bei der man vor der Aufnahme einen Bereich festlegen kann, der aufgenommen werden soll. Der aufzunehmende Bereich wird über einen frei skalierbaren, roten Rahmen festgelegt (siehe Abbildung 18).

Falls man mehrere Monitore angeschlossen haben sollte und dann die „Vollbildaufnahme“ auswählt, so kann man außerdem auswählen, welchen Bildschirm man aufnehmen möchte (siehe Abbildung 17).

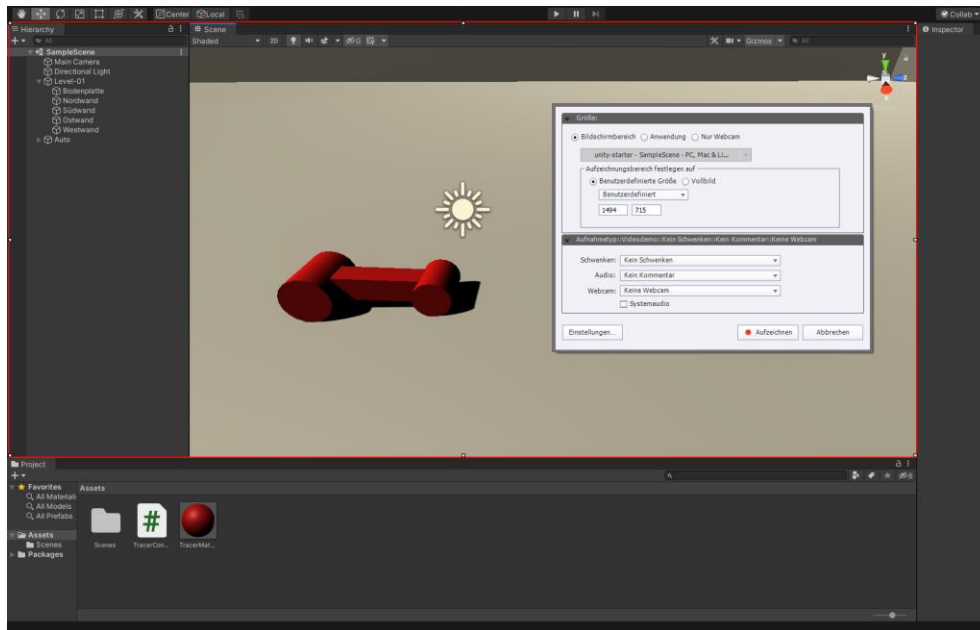


Abbildung 18: Screenshot aus Adobe Captivate 2019: Videodemo Voreinstellungen Bildschirmbereich: „Benutzerdefinierte Größe“

### Anwendung

Wählt man als Quelle für die Aufzeichnung „Anwendung“ aus, so kann man aus einer Dropdown-Leiste mit sämtlichen Anwendungen, die gerade auf dem PC geöffnet sind, auswählen, welche Anwendung man aufzeichnen möchte.

Hat man dann die gewünschte Anwendung ausgewählt (in unserem Fall Unity), so kann man wiederum zwischen „Anwendungsfenster“, „Anwendungsbereich“ und „Benutzerdefinierte Größe“ wählen.

Wählt man „Anwendungsfenster“ aus, so wird – wie der Name schon sagt – das gesamte Anwendungsfenster aufgenommen.

Wählt man „Anwendungsbereich“ aus, so kann man eine bestimmte Region in der Anwendung auswählen. In unserem Fall haben wir Unity als Anwendung ausgewählt. Unity besteht aus mehreren Anwendungsbereichen wie beispielsweise *Hierarchy*, *Scene*, *Inspector*, etc. Diese Bereiche werden automatisch von Captivate erkannt. Der rote Rahmen, der den Ausschnitt anzeigt, der aufgenommen werden soll, muss also nicht händisch gezogen werden, sondern legt sich automatisch um den Anwendungsbereich, der gerade mit der Maus fokussiert wird.

Wählt man „Benutzerdefinierte Größe“ aus, so kann man wieder über einen frei skalierbaren, roten Rahmen einen Bereich festlegen, der aufgenommen werden soll.

### Weitere Voreinstellungen

Zusätzlich zu den oben genannten Einstellungen über die Quelle, die aufgenommen werden soll, gibt es noch weitere Einstellungsmöglichkeiten.

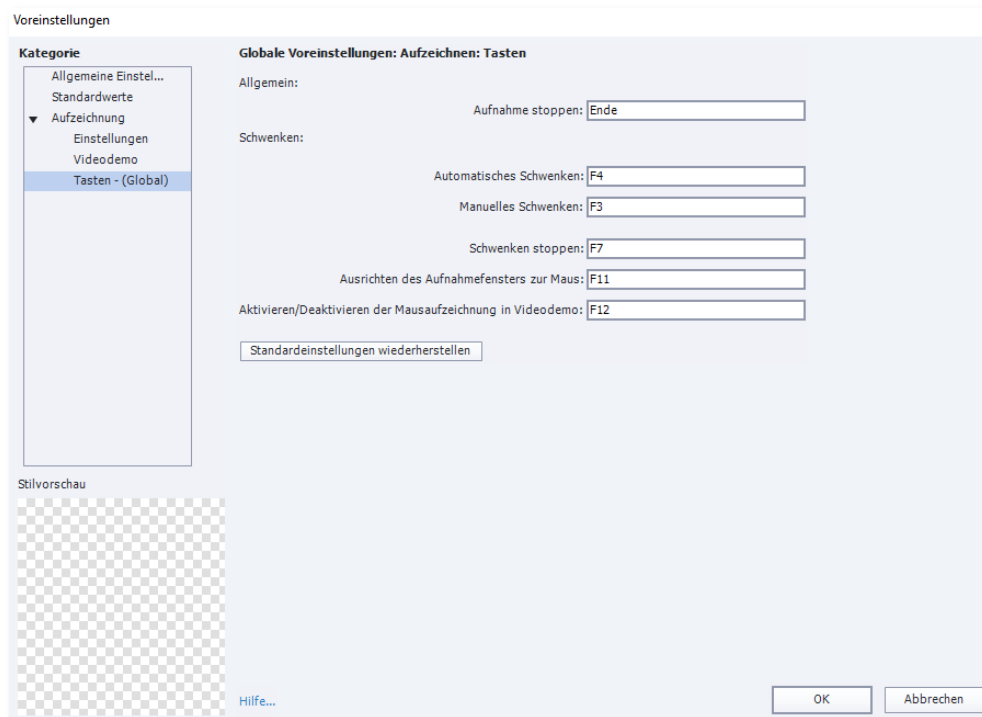


Abbildung 19: Screenshot aus Adobe Captivate 2019: Videodemo erweiterte Voreinstellungen

In diesen erweiterten Einstellungen kann man unter anderem auch festlegen, mit welchen Tastenkombinationen bestimmte Aktionen während der Aufnahme aufgerufen werden können. So kann man beispielsweise während der Aufnahme die F11-Taste drücken, wodurch zur Maus gezoomt (und somit der Fokus des Anwenders darauf gerichtet) wird. Das ist vor allem bei sehr kleinen, unübersichtlichen Schaltflächen sehr praktisch. Der Anwender sieht dadurch, wo sich der Mauszeiger gerade befindet und was genau angeklickt wird.

Im Rahmen dieser Arbeit ist es am sinnvollsten, den Bildschirm als Vollbildaufnahme aufzunehmen, da das Programm im Vollbildmodus verwendet werden soll und zu jeder Zeit das gesamte Programm sichtbar sein soll. In den Voreinstellungen muss dementsprechend also „Bildschirmbereich“ > „Vollbild“ > „Monitor 1“ ausgewählt werden.

## Screencast bearbeiten

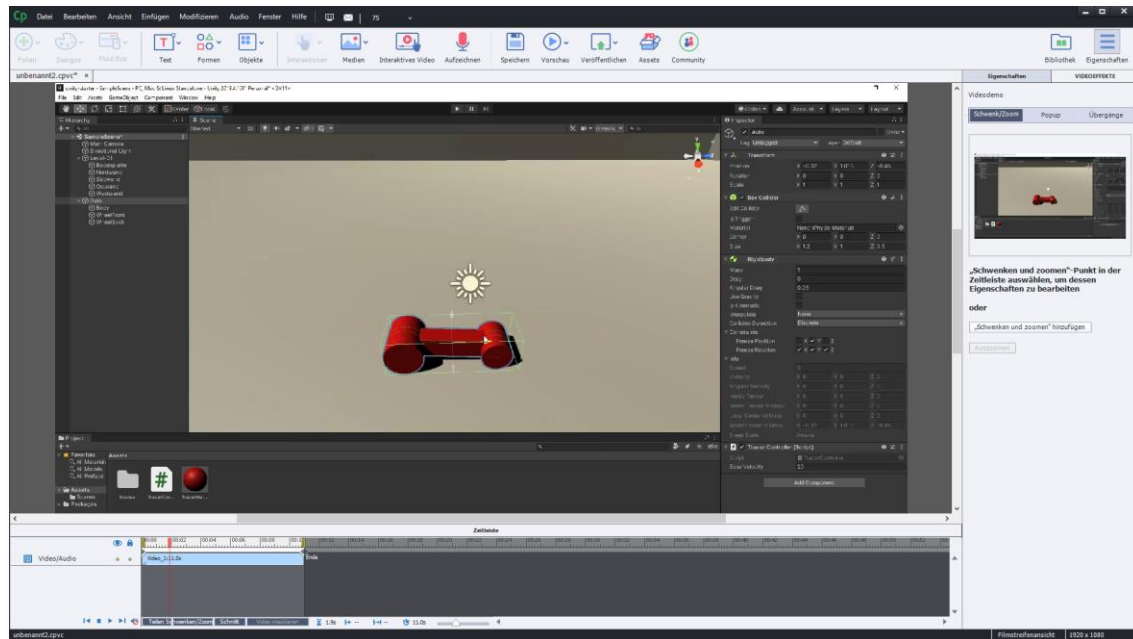


Abbildung 20: Screenshot aus Adobe Captivate 2019: Screencast bearbeiten

Sobald man die Bildschirmaufnahme beendet, erscheint die Aufnahme in diesem Fenster (siehe Abbildung 20). Hier kann die Aufnahme direkt in Adobe Captivate zugeschnitten, skaliert und mit Effekten geschmückt werden. So kann man zum Beispiel einen Zeitpunkt in der Zeitleiste auswählen und an dieser Stelle den Effekt „Schwenken und zoomen“ hinzufügen. Dieser Effekt kann also nicht nur während der Aufnahme über eine Tastenkombination benutzt werden, sondern auch im Nachhinein in der Bearbeitungsansicht.

## Screencast veröffentlichen

Wenn der Screencast fertig zugeschnitten und bearbeitet wurde, kann man das Video veröffentlichen. Hierfür gibt es wieder einige Voreinstellungen, die vorab festgelegt werden können (siehe Abbildung 21).

Hier können Audioeinstellungen festgelegt werden, Symbole ein- und ausgeblendet werden, Metadaten wie Projekt-Titel, Projekt-Autor, Unternehmen etc. eingetragen werden und vieles mehr.

Voreinstellungen

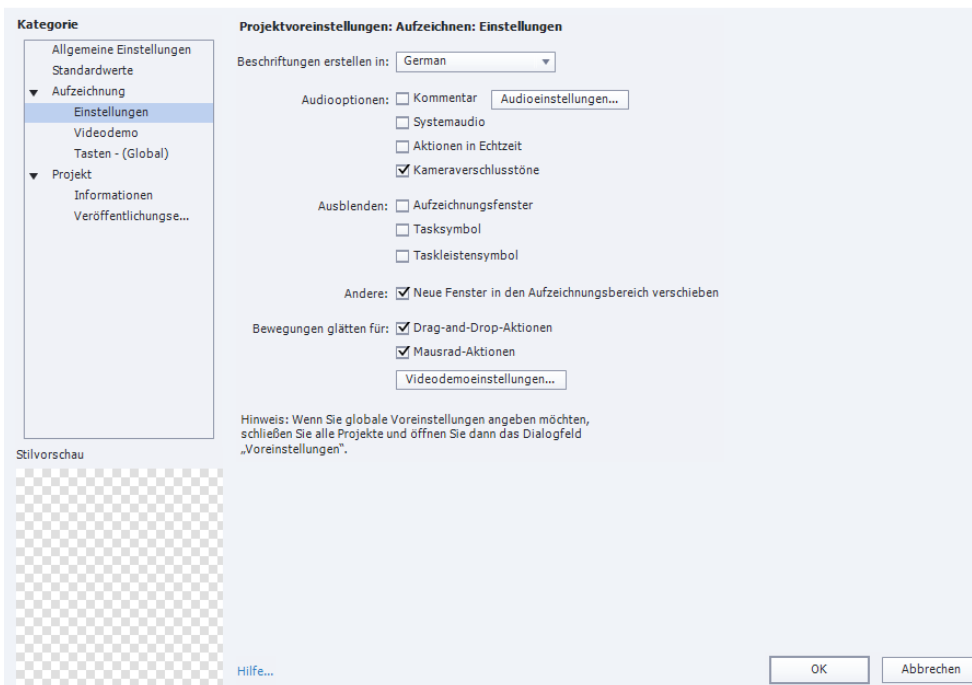


Abbildung 21: Screenshot aus Adobe Captivate 2019: Voreinstellungen zur Veröffentlichung

## Screencast mit GeForce Experience

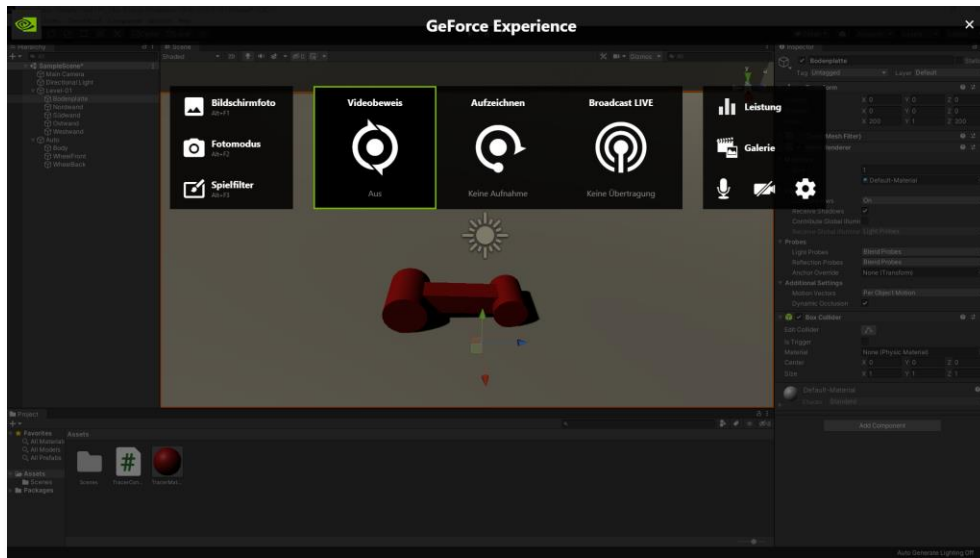


Abbildung 22: Screenshot von GeForce Experience Overlay

Nvidia GeForce Experience ist ein Grafikkarten-Tool für Nvidia-Grafikkarten. Wer eine Nvidia-Grafikkarte in seinem PC verbaut und Windows als Betriebssystem hat, kann GeForce Experience nutzen. Das Tool wird bei neueren Treiber-Versionen automatisch mitgeliefert, es kann aber auch jederzeit kostenlos von der Nvidia-Homepage heruntergeladen werden.

Die Funktion der Bildschirmaufnahme ist bei diesem Tool eigentlich vorrangig für Gaming ausgelegt, kann aber auch problemlos verwendet werden, um den Desktop und die Mausbewegungen aufzunehmen.

Über die Tastenkombination Alt + Z kann das GeForce Experience Overlay eingeblendet werden. Hier gibt es verschiedene Einstellungsmöglichkeiten für die Bildschirmaufnahme.

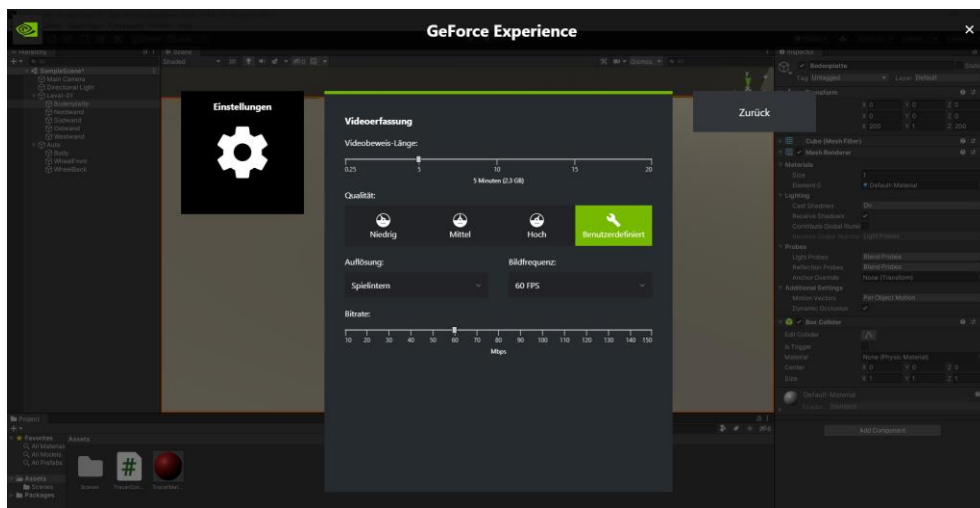


Abbildung 23: Screenshot von GeForce Experience Videoeinstellungen

Die Bildschirmaufnahme über GeForce Experience bietet gegenüber der Bildschirmaufnahme mit Adobe Captivate einige Vorteile:

- Kostenlos, wenn man eine Nvidia Grafikkarte verbaut und Windows als Betriebssystem hat
- Sehr einfache Handhabung – man kann die Bildschirmaufnahme in jeder Situation einfach starten und beenden, ohne zuvor eine Software öffnen zu müssen, ein neues Projekt anlegen zu müssen oder diverse Voreinstellungen festlegen zu müssen.
- Sehr gute, hochauflösende Qualität der Aufnahmen

Nachteile von GeForce Experience gegenüber Adobe Captivate:

- Kein implementiertes Tool für die anschließende Bearbeitung der Aufnahme
- Keine Zoom-Funktion während der Aufnahme
- Keine Möglichkeit nur einen Bildausschnitt aufzunehmen. Es wird immer der gesamte Bildschirm aufgenommen.

## **Praktische Durchführung: Erstellung der Lernvideos für das geplante E-Learning**

### *Screencast mit Adobe Captivate*

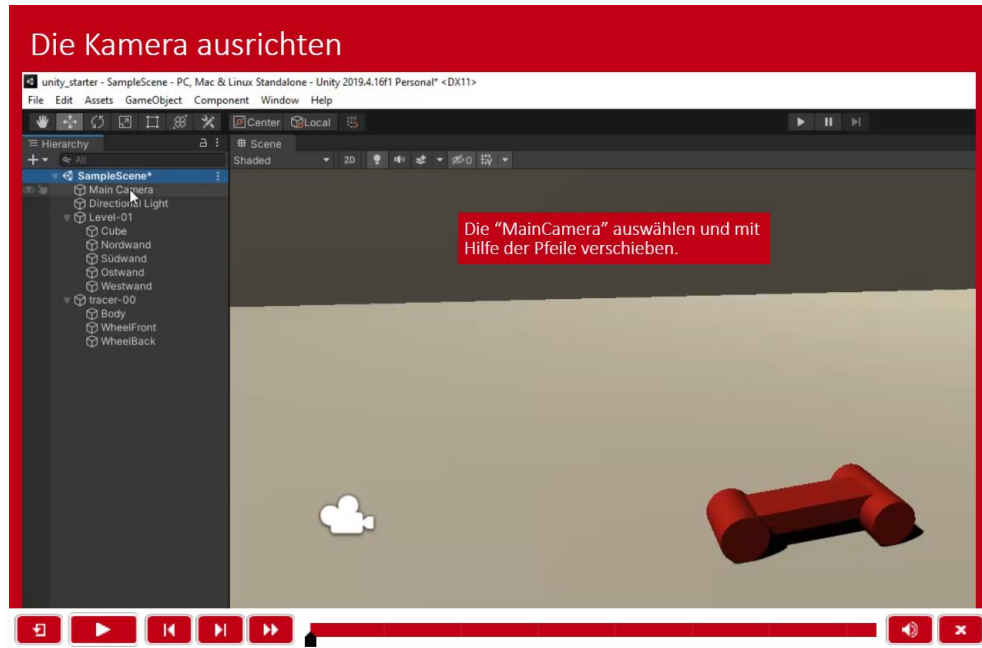
Die erste Bildschirmaufnahme, die im Rahmen dieser Arbeit aufgenommen wurde, war eine Aufnahme zu den Grundfunktionen von Unity, welche im Dokument `Unity_Einstieg.pdf` erläutert werden.

Wie zuvor geplant habe ich zunächst die zu zeigende Situation „geübt“. Es handelt sich um eine Bildschirmaufnahme für eine beschreibende Folie, in der gezeigt werden soll, wie man das Rotations-Tool auswählt und die Kamera mithilfe der Rotationslinien rotieren kann.

In den Voreinstellungen für die Bildschirmaufnahme habe ich als aufzunehmenden Bereich den „Bildschirmbereich“ als „Vollbild“ festgelegt und die Audioaufnahme deaktiviert.

Anschließend habe ich die Aufnahme gestartet, in Unity die Kamera und das Rotations-Tool ausgewählt und anhand des fertig modellierten Tracers, welcher in der ersten Übungsaufgabe erstellt werden soll, das Rotieren der Kamera mit dem Rotations-Tool demonstriert.

Nach dem Beenden der Aufnahme habe ich das Video in Captivate bearbeitet. Die Möglichkeiten sind hier ziemlich begrenzt – so habe ich lediglich ein paar Sekunden vom Video-Ende weggeschnitten und den Zoom-Effekt getestet.



**Abbildung 24:** Screenshot von Screencast mit Captivate: Test des Zoom-Effekts

Der Zoom-Effekt an sich ist sehr praktisch und fokussiert sehr gut die Aufmerksamkeit des Anwenders an die richtige Stelle. Allerdings hat die Zoom-Funktion in der aktuellen Captivate-Version (2019) einen Bug. Es wird in einer gewissen Geschwindigkeit, die man anpassen kann, hereingezoomt – allerdings wird anschließend nicht wieder herausgezoomt, sondern es wird abrupt in die ursprüngliche Vollbild-Ansicht zurückgesprungen. Da dies leider ein sehr harter und unschöner Übergang ist und es keine Möglichkeit gibt, das Problem in Captivate zu beheben, habe ich die Zoom-Funktion nur für diese eine Bildschirmaufnahme verwendet (siehe „1 Unity Einstieg.cptx“ Folie 2: Die Kamera ausrichten).

Anschließend habe ich die Aufnahme als MP4-Datei exportiert.

Der gesamte Prozess bis das Video fertig und für das E-Learning importierbar war, hat insgesamt deutlich länger gedauert als ursprünglich gedacht. Das Vorbereiten der Aufnahme (Überlegen, was gezeigt werden soll, das Szenario üben, die Voreinstellungen für die Aufnahme festlegen) und das anschließende Bearbeiten der Aufnahme nahmen sehr viel Zeit in Anspruch. Insgesamt saß ich an dieser einen Aufnahme ca. 1,5 Stunden. Und das Ergebnis aller Mühen war ein 26-Sekunden-langes Video.

Spätestens jetzt wusste ich, dass ich in dem mir zur Verfügung stehenden Bearbeitungszeitraum niemals zu allen Übungsaufgaben aus den zur Verfügung gestellten Vorlesungsunterlagen E-Learnings erstellen kann.

### *Screencast mit GeForce Experience*

Die erste Bildschirmaufnahme mit GeForce Experience habe ich bei der ersten Übungsaufgabe aus Unity\_Einstieg.pdf getestet.

Wie schon in Kapitel 3.3 „Softwarewissen umsetzen“ bereits gesagt, sollte man die Übungsaufgabe vor der Bildschirmaufnahme zunächst mindestens ein Mal üben. Das habe ich dann auch getan. Sämtliche Stolperstellen, an denen ich im ersten Durchlauf hängengeblieben bin, sind in Kapitel 4.2 „Ist-Analyse der Vorlesungsunterlagen“ zu finden.

Als ich die komplette Übung dann soweit durchgespielt hatte, dass ich alle schwierigen Stellen kannte, habe ich mich an die Bildschirmaufnahme gewagt. Hier habe ich mir zunächst überlegt, wie lang die jeweiligen Video-Sequenzen sein sollen und habe vor jeder Bildschirmaufnahme den zu demonstrierenden Vorgang nochmals geübt, anschließend alles wieder rückgängig gemacht und erst dann die Aufnahme gestartet. Dennoch musste ich die erste Aufnahme (und auch einige darauffolgende Aufnahmen) mehrmals neu starten.

Ich merkte jedoch schon bei der ersten Aufnahme mit GeForce Experience, dass mir diese Screencast-Methode schon deutlich besser gefiel. Da man im Vorherein nicht so viel einstellen musste, ging die Bildschirmaufnahme hier deutlich schneller von Statten. Wenn man sich dann doch mal verklickt hatte oder es sonstige Probleme bei der Aufnahme gab, so konnte man die Aufnahme einfach mit nur einer Tastenkombination stoppen und neustarten.

Nach Beenden der Aufnahme befand sich das Video dann direkt in perfekter Auflösung im MP4-Format in einem zuvor von mir zugewiesenen Ordner. Dieses Video musste aber natürlich so wie auch die Bildschirmaufnahme per Captivate zunächst noch bearbeitet werden.

Dafür kam dann erstmals das Programm Adobe Premiere Pro zum Einsatz. Dazu mehr im folgenden Kapitel.

### *Videos mit Adobe Premiere Pro nachbearbeiten und schneiden*

Um die erstellten Bildschirmaufnahmen mit Adobe Premiere Pro bearbeiten zu können, muss zunächst ein neues Premiere-Projekt angelegt werden.

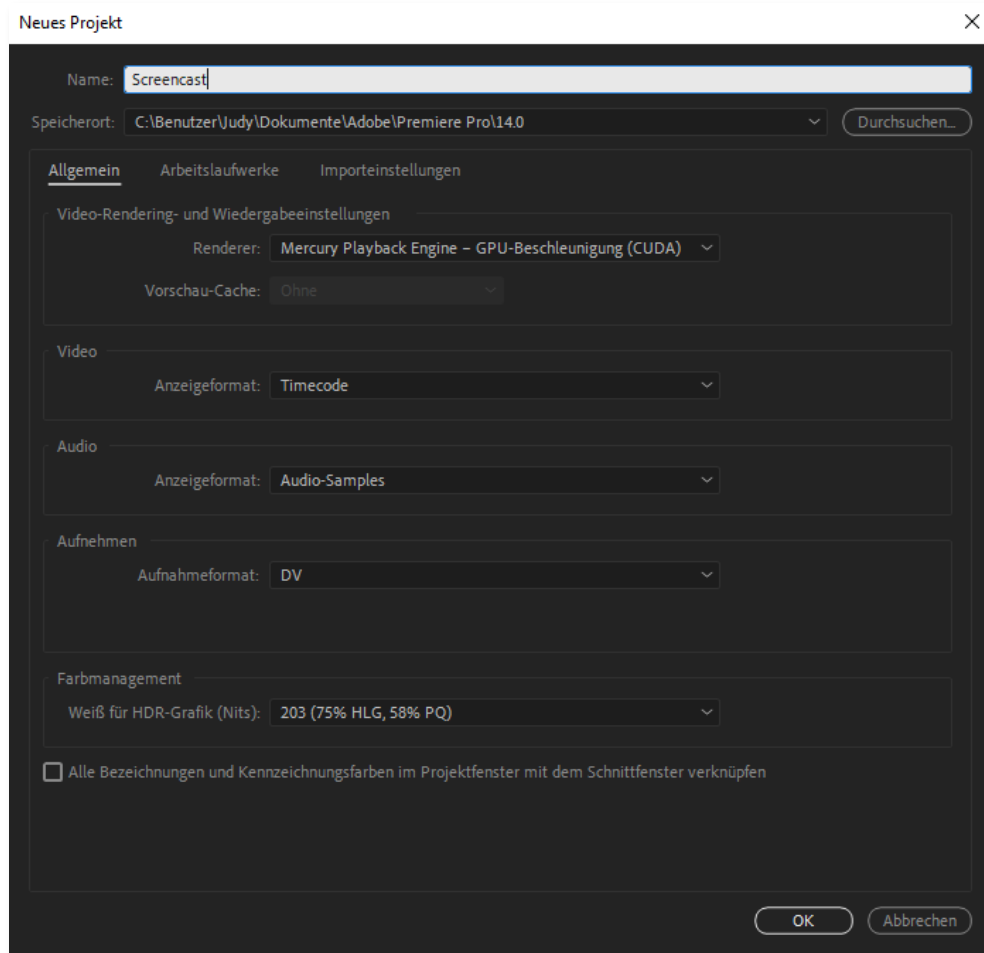
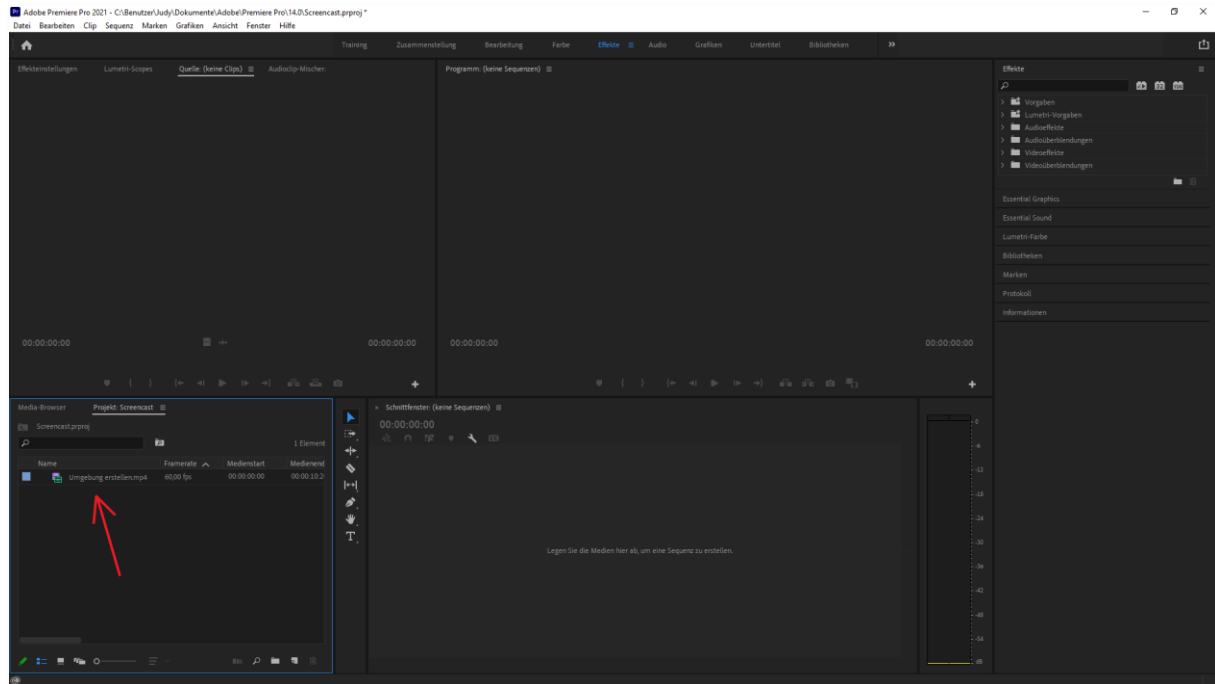


Abbildung 25: Screenshot von Adobe Premiere Pro: Ein neues Projekt anlegen

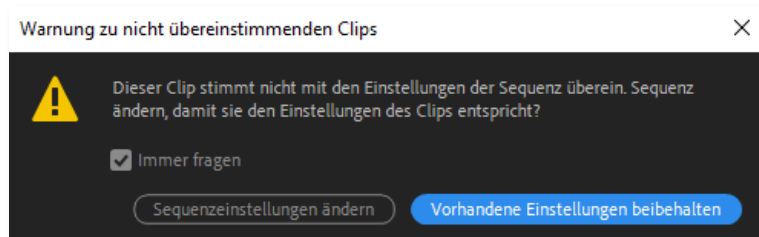
Bei den Projekteinstellungen muss nichts weiter beachtet werden. Die Projekteinstellungen werden sich im Nachhinein mit nur einem Klick an die Einstellungen der Bildschirmaufnahme anpassen lassen.

Wenn man das Projekt angelegt hat, können im nächsten Schritt die Videodateien in den *Media-Browser* importiert werden. Das muss man machen, da man die Videodateien ansonsten nicht im *Schnittfenster* bearbeiten kann.



**Abbildung 26:** Screenshot von Adobe Premiere Pro: Bildschirmaufnahme in den Media-Browser importieren

Sobald die Datei(en) importiert wurde(n), kann die Aufnahme per Drag-and-Drop in das Schnittfenster gezogen werden. Es wird nun automatisch eine Warnung erscheinen, in der steht, dass der Clip nicht mit den Sequenzeinstellungen übereinstimmt (siehe Abbildung 27). Hier kann man nun auf „Sequenzeinstellungen ändern“ klicken und die Sequenzeinstellungen werden sich automatisch der Bildschirmaufnahme anpassen.



**Abbildung 27:** Screenshot von Adobe Premiere Pro: Warnung zu nicht übereinstimmenden Clips

Anschließend kann der Clip bearbeitet werden. Falls die Bildschirmaufnahme eine Tonspur haben sollte, so kann man diese entfernen, indem man auf den Clip rechtsklickt und anschließend „Verknüpfung aufheben“ auswählt (siehe Abbildung 28).

Danach kann man die Tonspur auswählen und diese mit der entf-Taste entfernen.

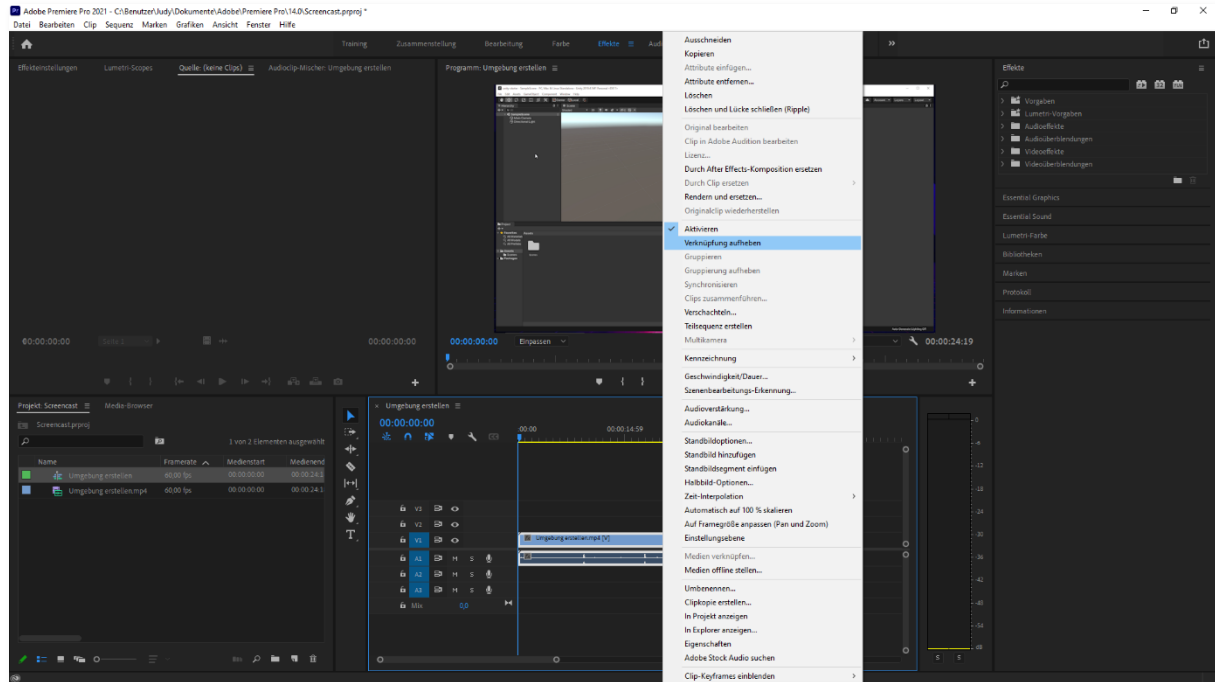


Abbildung 28: Screenshot von Adobe Premiere Pro: Audio entfernen durch „Verknüpfung aufheben“

Nun kann der Clip beliebig zugeschnitten werden. Hierbei habe ich sämtliche Stellen, an denen die Maus länger als 2 Sekunden nicht bewegt wird, herausgeschnitten, um einen angenehmen Videofluss ohne Unterbrechungen zu gewährleisten. Wenn man bei der Bildschirmaufnahme also mal während der Übung kurz stehenbleiben sollte, und nicht mehr weiß, was man als nächstes tun muss, dann ist das kein großes Problem. Man kann die Maus einfach da lassen, wo sie ist und in Ruhe überlegen, welcher Schritt als nächstes folgt. Solche Pausen lassen sich mit Premiere Pro schnell und einfach aus der Aufnahme herauschneiden.

Um einen schönen Übergang zwischen den Folien-Videos zu ermöglichen, habe ich außerdem ab dem zweiten E-Learning bei sämtlichen Bildschirmaufnahmen am Ende den Blende-Effekt „Übergang zu Schwarz“ hinzugefügt. Mit diesem Effekt endet das Video nicht abrupt, sondern es gibt, wie der Name schon sagt, einen schönen, sanften Übergang zu einem schwarzen Bildschirm.

Man kann diesen Effekt ganz einfach in der Effekte-Liste auswählen und dann per Drag-and-Drop auf das Ende der Sequenz ziehen (siehe Abbildung 29). *Effekte > Videoüberblendungen > Blende > Übergang zu Schwarz*

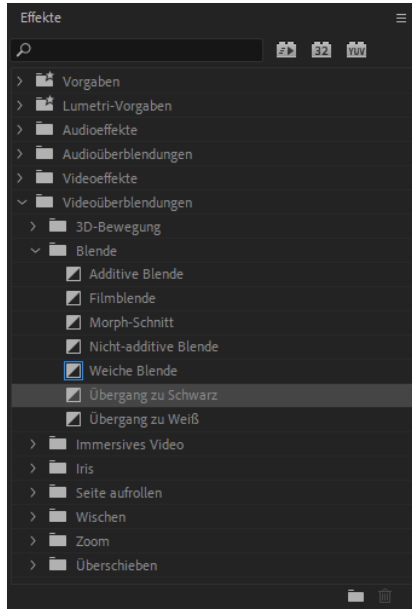


Abbildung 29: Screenshot von Adobe Premiere Pro: Blende-Effekt „Übergang zu Schwarz“

Wenn die Sequenz dann fertig bearbeitet wurde, kann man die Sequenz markieren und über *Datei > Exportieren > Medien...* exportieren. Auch hier können wieder einige Voreinstellungen festgelegt werden:

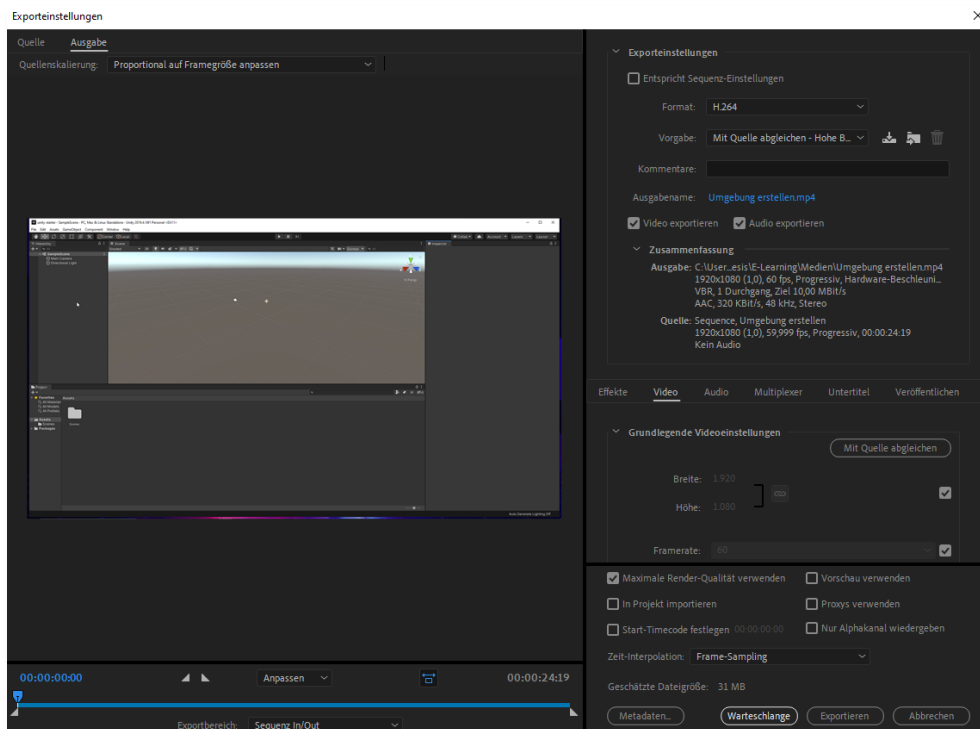


Abbildung 30: Screenshot von Adobe Premiere Pro: Exporteinstellungen

Als Ausgabeformat muss hier „H.264“ ausgewählt werden. Das Ergebnis nach dem Export ist eine MP4-Datei. Des Weiteren kann man hier direkt einen geeigneten Namen für die Aufnahme festlegen.

Wir möchten für unsere Bildschirmaufnahme außerdem die bestmögliche Qualität haben, da es sich um eine Bildschirmaufnahme von einer Systemoberfläche handelt. Es muss daher zu jederzeit die Schrift sämtlicher Schaltflächen lesbar und die Position des Mauszeigers gut sichtbar sein. Wenn der Anwender aufgrund von niedriger Video-Qualität nicht erkennen kann, was in der Bildschirmaufnahme eingegeben bzw. angeklickt wird, dann ist das sehr schlecht. Dann würde das gesamte E-Learning seinen Zweck nicht erfüllen. Daher sollte das Kontrollkästchen bei „*Maximale Render-Qualität verwenden*“ aktiviert sein.

Nach dem Export ist die Bildschirmaufnahme bereit, in einem E-Learning als Folien-Video verwendet zu werden.

## E-Learning mit Adobe Captivate erstellen

### Ein neues Projekt anlegen

Sobald die Lernvideos für das E-Learning zugeschnitten, mit Effekten versehen und exportiert wurden, kann in Captivate ein neues Projekt für das geplante E-Learning angelegt werden.



Abbildung 31: Screenshot von Adobe Captivate: Ein neues Projekt anlegen

Hier wählt man am besten das „Projekt mit automatischer Anpassung“ aus. Es bedeutet, dass das E-Learning sich nach der Veröffentlichung automatisch an die Bildschirmgröße des Ausgabegeräts anpasst (Handy, Tablet, Monitor, ...).

Sobald man auf „Erstellen“ geklickt hat, öffnet sich das neu erstellte Projekt. Dieses Projekt ist zu Anfang logischerweise komplett leer und es ist noch kein Design festgelegt. Standardmäßig sind die Folien des Projekts weiß.

## Die Designfarbe ändern

In diesem Fall soll das E-Learning aber keine weißen Folien haben, sondern es soll schon am Design erkennbar sein, dass es sich um ein E-Learning aus der Multimedia-Vorlesung handelt. Daher habe ich mich beim Design für einen roten Projekthintergrund entschieden, so wie er auch in sämtlichen Vorlesungsunterlagen jeweils auf der ersten Folie verwendet wird.

Um ein benutzerdefiniertes Design für die Folien festzulegen, muss man in Captivate in der oberen Leiste auf *Designs > Designfarben* klicken.

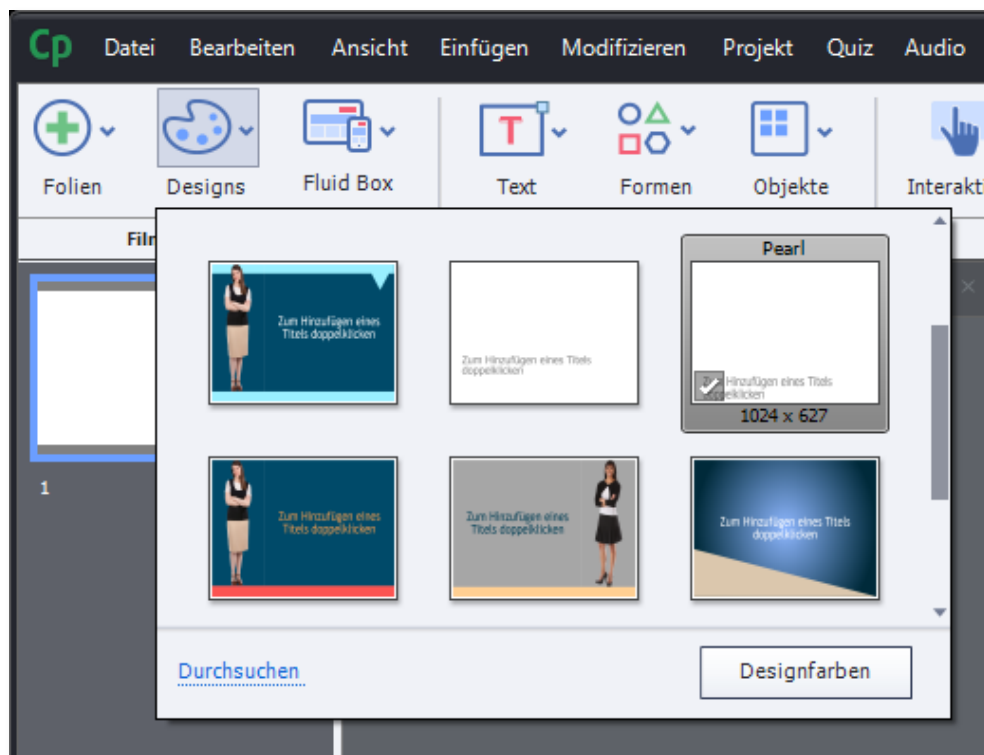


Abbildung 32: Screenshot von Adobe Captivate: Designfarben ändern

Es öffnet sich ein weiteres Fenster, in dem aus einem Dropdown-Menü zwischen vielen verschiedenen vorgefertigten Design-Kompositionen ausgewählt werden kann (siehe Abbildung 33). In diesem Fall habe ich das Design „Chic\_Shade“ ausgewählt, bei dem die Folien direkt im gewünschten Rotton eingefärbt werden.

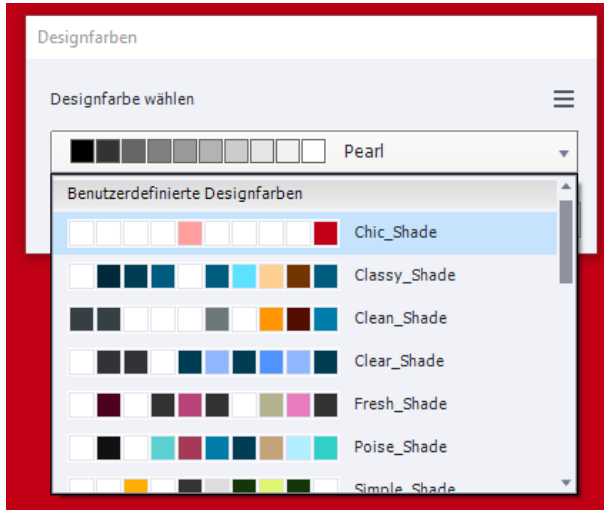


Abbildung 33: Screenshot von Adobe Captivate: Designfarbe auswählen

Sobald man die gewünschte Designfarbe ausgewählt hat, kann man für die ausgewählte Designfarbe bei Bedarf weitere Anpassungen vornehmen.

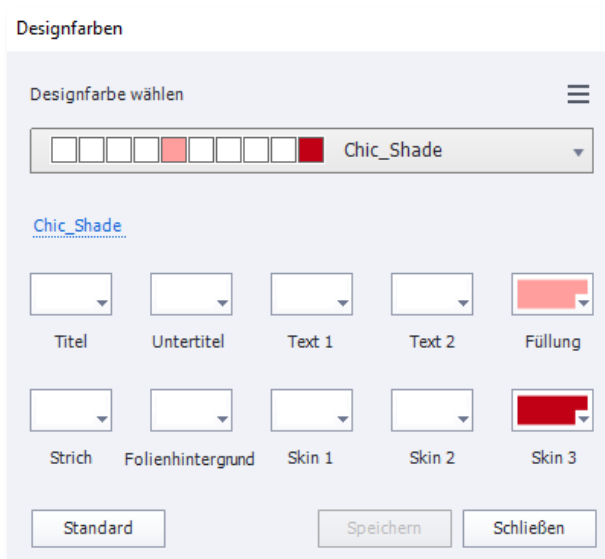


Abbildung 34: Screenshot von Adobe Captivate: Designfarbe anpassen

## Textfelder einfügen

Um die Folie nun mit einer aussagekräftigen Überschrift ausstatten zu können, muss man auf *Text > Textbeschriftung* klicken.

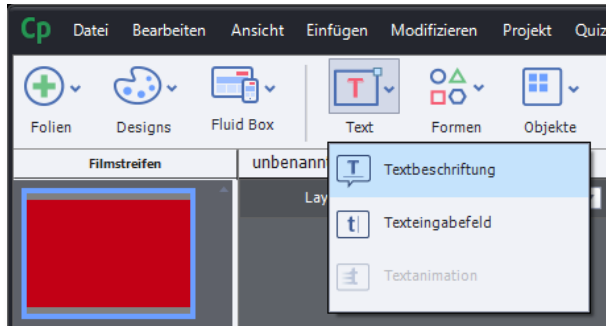


Abbildung 35: Screenshot von Adobe Captivate: Ein Textfeld einfügen

Nun öffnet sich automatisch ein Textfeld auf der Folie, in das ein beliebiger Text eingegeben werden kann. Dieser Text kann im *Eigenschaften*-Fenster beliebig angepasst werden (Schriftart, Auszeichnung, Schriftgröße, rechtsbündig/zentriert/linksbündig/Blocksatz, Hintergrundfarbe des Textfeldes und mehr, siehe Abbildung 36).

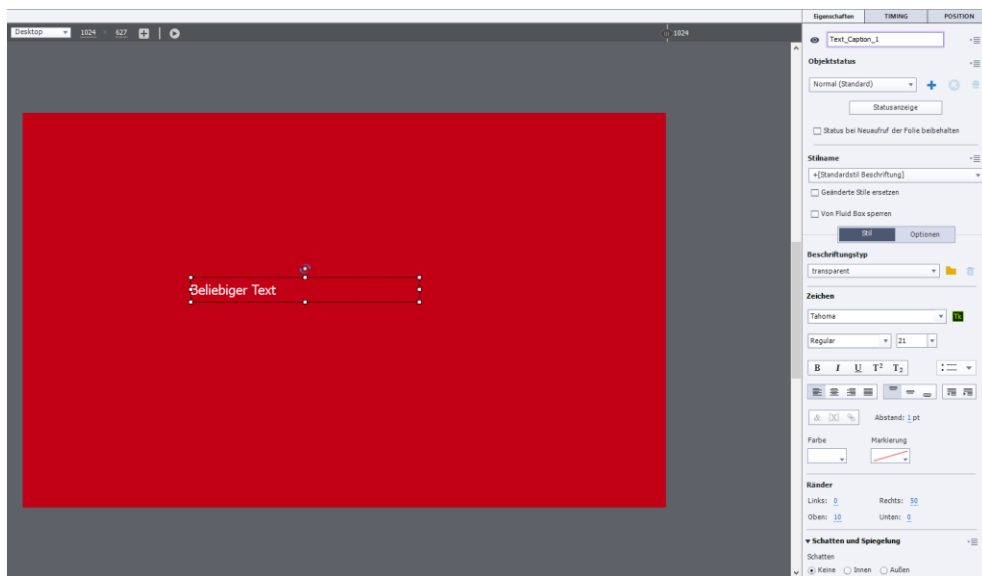


Abbildung 36: Screenshot von Adobe Captivate: Texteeigenschaften über Eigenschaften-Fenster anpassen

## Ein Folienvideo einfügen

Um ein Folienvideo einzufügen, muss man auf *Medien* > *Video* klicken und in dem Fenster, das sich daraufhin öffnet, „Folienvideo“ auswählen. Außerdem sollte „Foliendauer an Video anpassen“ ausgewählt sein. Anschließend kann man über die „Durchsuchen“-Schaltfläche zu der Video-Datei navigieren, die eingefügt werden soll (siehe Abbildung 37). Spätestens jetzt wird man froh sein, wenn man den Lernvideos sprechende Namen gegeben hat, sodass man nun ohne große Probleme das richtige Video auswählen kann.

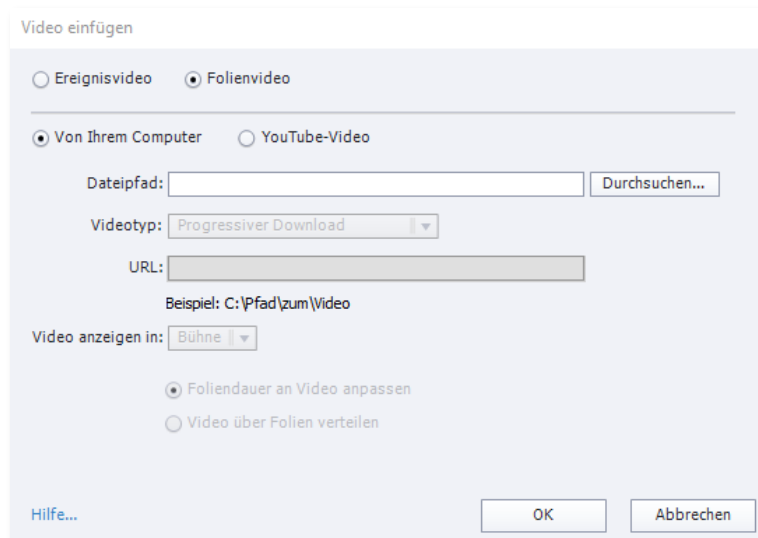


Abbildung 37: Screenshot von Adobe Captivate: Ein Folienvideo einfügen

Sobald das Folienvideo eingefügt wurde, kann dieses entsprechend der Foliengröße skaliert und positioniert werden. Wenn man während dem Skalieren zusätzlich die shift-Taste gedrückt hält, dann werden alle Seiten des Videos proportional skaliert.

## Anzeigedauer der Elemente in der Zeitleiste festlegen

Damit das zuvor erstellte Textfeld mit der Überschrift der Folie auch während der gesamten Dauer des Folienvideos eingeblendet bleibt, muss die Anzeigedauer des Textfeldes in der „Zeitleiste“ festgelegt werden. Wenn man dies nicht tut, dann kann es passieren, dass das Textfeld beispielsweise nur 3 Sekunden zu sehen ist und dann ausgeblendet wird (siehe Abbildung 38).

Die Anzeigedauer kann ganz einfach angepasst werden, indem man in der Zeitleiste beim Textfeld das Ende der Anzeigedauer-Schaltfläche greift und bis an das Ende des Folienvideos zieht (siehe Abbildung 39).

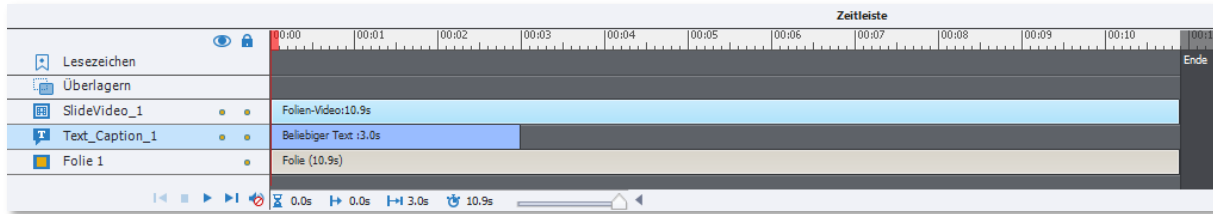


Abbildung 38: Screenshot von Adobe Captivate: Anzeigedauer des Textfeldes in der Zeitleiste anpassen (vor der Anpassung)

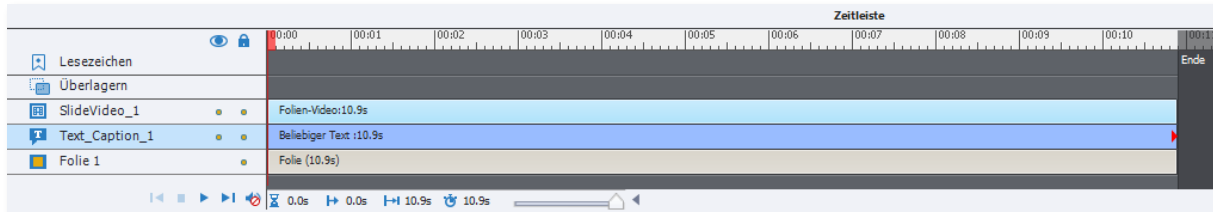


Abbildung 39: Screenshot von Adobe Captivate: Anzeigedauer des Textfeldes in der Zeitleiste anpassen (nach der Anpassung)

Genau so kann man auch Elemente wie weitere Textfelder, Pfeile, Grafiken und vieles mehr an geeigneter Stelle einblenden und auch wieder ausblenden lassen. So habe ich beispielsweise im Ersten E-Learning „1 Unity Einstieg“ Auf Folie 17 jeweils Bilder für die Tasten QWEASD einblenden lassen, die in genau dem Moment, in dem sie während der Bildschirmaufnahme gedrückt wurden, rot aufleuchten.

Effekte dieser Art sind sehr schön anzusehen und tragen viel zum besseren Verständnis des Lernvideos bei, allerdings ist der Aufbau einer solchen Folie sehr zeitaufwendig und mit sehr viel „friemelei“ verbunden (siehe Abbildung 40). Hier muss man je nach Situation abwägen, ob der Aufwand das Ergebnis wert ist, oder ob man es lieber einfach halten möchte.

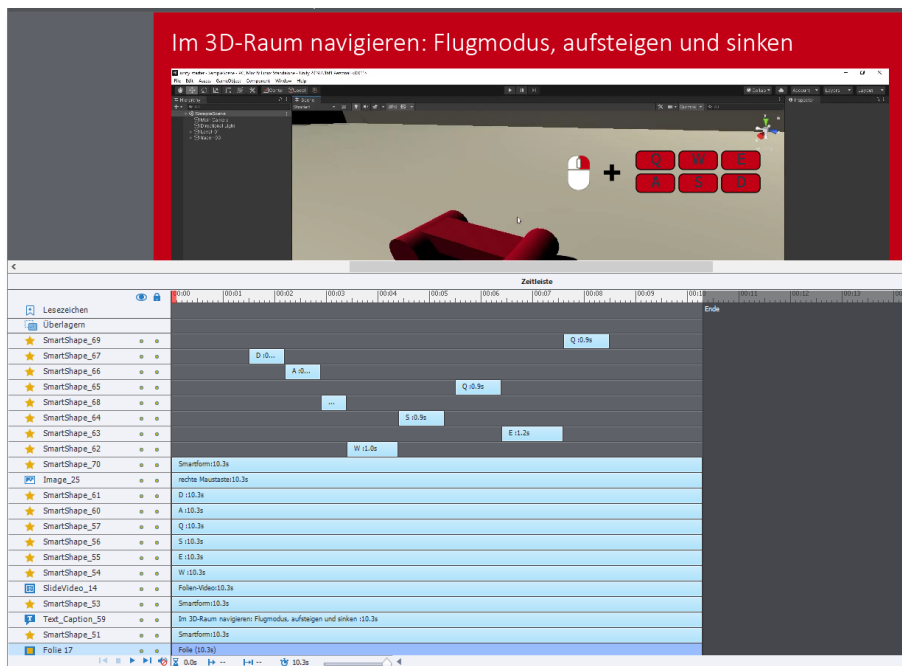


Abbildung 40: Screenshot von E-Learning „1 Unity Einstieg“ auf Folie 17

Ich wollte für das E-Learning Mauszeiger-Icons haben, die zeigen, welche Maustaste geklickt, bzw. gedrückt gehalten werden soll. Als ich keine kostenlosen Icons finden konnte, die meiner Vorstellung entsprachen, habe ich mit Adobe Illustrator kurzerhand eigene Icons erstellt und die Taste, die jeweils geklickt werden muss, in genau dem Rot eingefärbt, das dem Rot der Folien-Designfarbe entspricht.

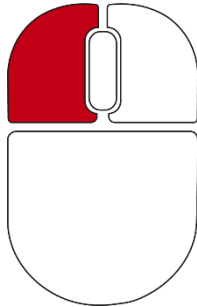


Abbildung 41: Icon linke Maustaste

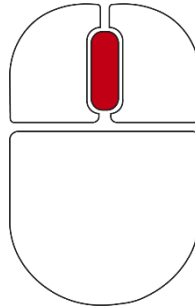


Abbildung 42: Icon Mausehrad

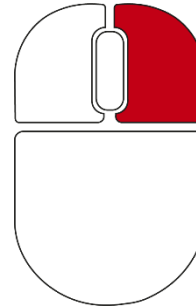


Abbildung 43: Icon rechte Maustaste

## Weitere Folien erstellen

Wenn dann bei allen Elementen, die auf der Folie angezeigt werden sollen, die korrekte Anzeigedauer eingestellt wurde und diese Folie somit fertig ist, kann die nächste Folie erstellt werden. Hierzu muss man auf *Folien > Leere Folie* klicken (siehe ).

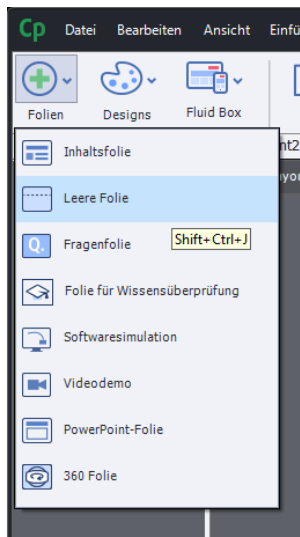


Abbildung 44: Screenshot von Adobe Captivate: Neue leere Folie hinzufügen

Diese Folie wird nun zunächst wieder als weiße Folie angezeigt werden. Das liegt daran, dass im *Eigenschaften*-Fenster als *Hintergrund* der „Projekthintergrund“ ausgewählt ist. Hier muss man aus dem Drop-Down-Menü lediglich den „Folienmasterhintergrund“ auswählen und schon hat die Folie wieder den gewünschten Farbton.

## Das Projekt veröffentlichen

Wenn alle für das E-Learning notwendigen Folien angelegt und fertig befüllt wurden, kann das Projekt veröffentlicht werden. Vor dem Veröffentlichen ist es jedoch ratsam, sich eine Vorschau des erstellten E-Learnings anzuschauen. Hierzu muss man auf *Vorschau > Projekt* klicken, woraufhin sich der Browser öffnet, in dem das E-Learning abgespielt werden kann. Oft fallen hier kleine Fehler auf, die man in Captivate selbst gar nicht gesehen hat. Beispielsweise Textfelder, die zu schnell wieder verschwinden oder auch Inhalte, die eigentlich gar nicht da sein sollten.

Außerdem sollten vor dem Veröffentlichen die Veröffentlichungseinstellungen festgelegt werden. Hierzu klickt man auf *Datei > Veröffentlichungseinstellungen...*

Hier können wie beim Veröffentlichen der Bildschirmaufnahme wieder sämtliche Einstellungen vorgenommen werden. Was man in jedem Fall beachten sollte, sind die Projekteinstellungen „*Informationen*“ und „*Größe und Qualität*“. In den Informationseinstellungen können sämtliche Metadaten wie Autor, Unternehmen und Projektname eingetragen werden. In den Qualitätseinstellungen kann, wie der Name schon vermuten lässt, die Qualität des E-Learnings festgelegt werden. Wie zuvor schonmal gesagt, soll für diese Arbeit die bestmögliche Qualität genommen werden. Daher sollte hier die Qualität auf „*Hoch*“ gesetzt werden (siehe Abbildung 45).

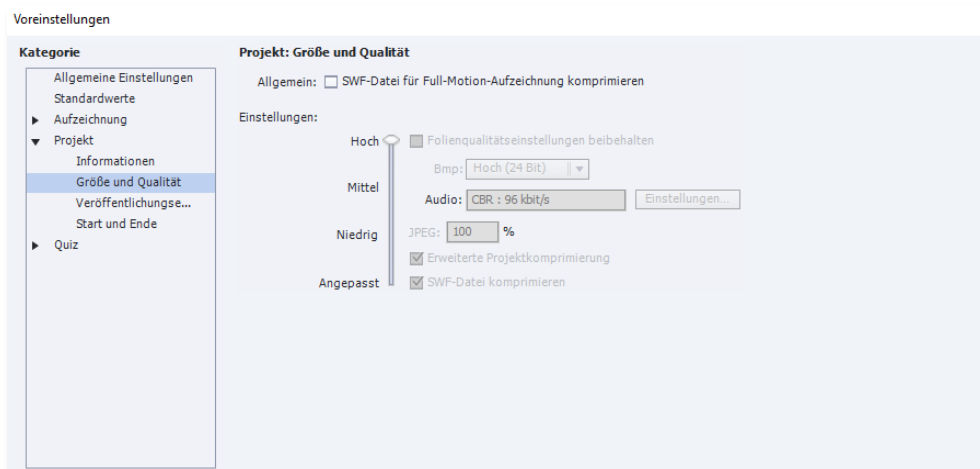


Abbildung 45: Screenshot von Adobe Captivate: Veröffentlichungseinstellungen

In den „*Veröffentlichungseinstellungen*“ kann dann außerdem auch die FPs-Rate hochgestellt werden. Das Maximum, das hier festgelegt werden kann, sind 50 Bilder pro Sekunde. Je höher die FPs-Rate, desto flüssiger laufen die Videos. Abschließend klickt man auf „*OK*“, um die Angaben zu speichern.

Zuletzt klickt man auf *Datei > Veröffentlichen...* Hier können der Projekttitle, der Speicherort und weitere Einstellungen festgelegt werden. Um das Projekt nun zu erstellen, muss man auf „*Veröffentlichen*“ klicken. Das fertige Projekt wird nach erfolgreicher Veröffentlichung automatisch im Browser geöffnet und kann direkt gestartet werden.

**Auf Geräten veröffentlichen**

Veröffentlichen als: HTML5

Projekttitle:

Speicherort:

Zip-Dateien

Typekit-Domänen:

Fügen Sie bis zu zehn Domänen hinzu, z. B. \*.adobe.com, localhost, 127.0.0.1

Folien: 2	Gesten für Mobilgeräte: <a href="#">Ja</a>
Folien mit Audio: 0	Geolocation: <a href="#">Nein</a>
Audioeinstellungen: <a href="#">Benutzerd...</a>	Barrierefreiheit: <a href="#">Ja</a>
Bewertung anzeigen: <a href="#">Nein</a>	eLearning-Ausgabe: <a href="#">Deaktiviert</a>

**Abbildung 46:** Screenshot von Adobe Captivate: Projekt veröffentlichen

## 5. Leitfaden für das Erstellen eines E-Learnings auf Basis von Vorlesungsunterlagen

### Ziel

Dieser Leitfaden soll Assistenten und Hiwis bei der Umsetzung von weiteren E-Learnings mit Lernvideos helfen. Es handelt sich hierbei um eine stark komprimierte Schritt-für-Schritt-Anleitung, die auf der Bachelor-Thesis „Entwicklung und Konzeption eines E-Learnings mit Lernvideos zu Virtual-Reality- und Augmented-Reality-Anwendungen auf Basis von Vorlesungsunterlagen“ basiert.

### Benötigte Software

Folgende Programme werden für die Erstellung von E-Learnings benötigt:

- Screencast: Nvidia GeForce Experience
- Videobearbeitung: Adobe Premiere Pro
- E-Learning: Adobe Captivate

### 1. Lernvideos per Screencast erstellen

#### Vorbereitung

1. Die Handlung, die aufgenommen werden soll, „üben“, sodass man sicher weiß, was zu tun ist.
2. Mit der Tastenkombination Alt + Z das GeForce Experience Overlay aufrufen
3. Unter *Einstellungen* > *Video* die maximale Qualität für die Bildschirmaufnahme auswählen
4. Bei *Einstellungen* > *Datenschutzkontrolle* > *Desktop-Erfassung* den Haken aktivieren
5. Das Overlay über die esc-Taste schließen.
6. Den Desktop aufräumen und alle irrelevanten Programme schließen.
7. Die Software auf dem Bildschirm so einrichten, dass alles, das gesehen werden soll, zu sehen ist.

#### Durchführung

1. Die Aufnahme starten durch die Tastenkombination alt + F9.
2. Die aufzuzeichnenden Handlungen durchführen.
3. Die Aufnahme beenden durch erneutes Drücken der Tastenkombination alt + F9
4. Diesen Vorgang beliebig oft wiederholen

## 2. Erstellte Videos zuschneiden und bearbeiten

1. Ein neues Projekt anlegen. Bei den Voreinstellungen können die Standardwerte eingestellt bleiben.
2. Die Bildschirmaufnahmen in Adobe Premiere Pro per Drag-and-Drop in den *Media-Browser* ziehen.
3. Das zu bearbeitende Video in das *Schnittfenster* ziehen. Falls die Warnung „*Warnung zu nicht übereinstimmenden Clips*“ erscheint, dann auf „*Sequenzeinstellungen ändern*“ klicken.

### *Unerwünschte Tonspur entfernen*

1. Rechtsklick auf die Sequenz im Schnittfenster > *Verknüpfung aufheben*
2. Die Audiospur im Schnittfenster auswählen.
3. Die Audiospur durch Drücken der entf-Taste löschen.

### *Inhalte herauschneiden*

1. Zum Startpunkt der Stelle navigieren, die herausgeschnitten werden soll.
2. Mit der Tastenkombination strg + K einen Cut einfügen.
3. Zum Endpunkt der Stelle navigieren, die herausgeschnitten werden soll.
4. Mit der Tastenkombination strg + K einen Cut einfügen.
5. Das nun freigestellte Segment anklicken und mit der entf-Taste löschen.
6. Rechtsklick in die Lücke und „*Löschen und Lücke schließen*“ auswählen.

### *Blende-Effekt hinzufügen*

1. Zu *Effekte > Videoüberblendungen > Blende > Übergang zu Schwarz* navigieren
2. Den Effekt per Drag-and-Drop auf das Ende und/oder den Anfang der Sequenz ziehen. Man kann den Effekt auch innerhalb der Sequenz verwenden, falls erwünscht.

### *Sequenz exportieren*

1. *Datei > Exportieren > Medien...*
2. *Ausgabeformat „H.264“ festlegen*
3. Einen sprechenden Namen vergeben
4. Bei „*Maximale Render-Qualität verwenden*“ das Häkchen setzen
5. Auf „*Exportieren*“ klicken

### 3. E-Learning erstellen

#### Neues Projekt anlegen

1. „Projekt mit automatischer Anpassung“ auswählen
2. Auf „Erstellen“ klicken

#### Designfarbe ändern

1. *Designs* > *Designfarben* auswählen
2. Im Dropdown-Menü eine Designfarbe auswählen
3. Die einzelnen Farben der Designfarbe ggf. anpassen

#### Textfelder einfügen

1. *Text* > *Textbeschriftung* auswählen
2. Einen beliebigen Text in das Textfeld eingeben
3. Den Text im *Eigenschaften*-Fenster beliebig formatieren

#### Folienvideo einfügen

1. *Medien* > *Video* auswählen
2. Folienvideo auswählen
3. Bei „*Foliendauer an Video anpassen*“ das Häkchen setzen
4. *Durchsuchen...* auswählen und die gewünschte Video-Datei auswählen
5. Auf „OK“ klicken
6. Das Folienvideo skalieren – während dem Skalieren die shift-Taste gedrückt halten, damit das Video proportional skaliert wird.
7. Das Folienvideo auf der Folie positionieren

#### Anzeigedauer der Elemente in der Zeitleiste festlegen

- Um den Zeitpunkt der Sichtbarkeit zu verschieben, ohne die Dauer der Sichtbarkeit zu verändern, die Objekt-Schaltfläche greifen und über die Zeitleiste ziehen.
- Um die Dauer der Sichtbarkeit anzupassen, den Anfang oder das Ende der Objekt-Schaltfläche greifen und bis zu der Stelle ziehen, bis zu der das Objekt sichtbar sein soll.

#### Weitere Folien hinzufügen

1. *Folien* > *Leere Folie* auswählen
2. Im *Eigenschaften*-Fenster als „*Hintergrund*“ den „*Folienmasterhintergrund*“ auswählen.

### *Projektvorschau anzeigen lassen*

- *Vorschau > Projekt* auswählen

### *Projekt veröffentlichen*

1. *Datei > Veröffentlichungseinstellungen...* auswählen
2. Bei *Projekt > Informationen* die Felder *Autor, Unternehmen, Projektname* etc. befüllen.
3. Bei *Projekt > Größe und Qualität* die Qualität auf *Hoch* setzen.
4. Bei *Projekt > Veröffentlichungseinstellungen* die *FPs-Rate* auf *50 Bilder pro Sekunde* setzen.
5. Die eingaben mit „OK“ bestätigen.
6. *Datei > Veröffentlichen...* auswählen
7. *Projekttitel* vergeben und *Speicherort* angeben
8. Auf „*Veröffentlichen*“ klicken

## 6. Abbildungsverzeichnis

Abbildung 1: Screenshot aus Unity_Einstieg.pdf – Die Programmregionen (Schober 2019:7).....	14
Abbildung 2: Screenshot aus Unity_Einstieg.pdf – Erstellen der Umgebung für Level-01(Schober 2019:19) .....	14
Abbildung 3: Screenshot aus Unity_Einstieg.pdf – Tracer einfärben (Schober 2019:24) .....	15
Abbildung 4: Screenshot aus Unity_Einstieg.pdf – Einen Box Collider für das Gefährt erstellen (Schober 2019:27).....	16
Abbildung 5: Ergebnis der Übung „Texte eingeben/ausgeben durch Mausklick“ .....	17
Abbildung 6: Screenshot aus Unity_CS_Texteingabe_Textausgabe.pdf – Texte eingeben/ausgeben durch Mausklick (Schober 2019:1).....	18
Abbildung 7: Screenshot aus Unity_CS_Texteingabe_Textausgabe.pdf – C#-Skript anlegen (Schober 2019:8).....	19
Abbildung 8: Screenshot aus Unity_CS_Texteingabe_Textausgabe.pdf – Das Skript schreiben (Schober 2019:10).....	19
Abbildung 9: Screenshot aus Unity_CS_Texteingabe_Textausgabe.pdf – Dem Button ein C#-Skript hinzufügen (Schober 2019:9).....	20
Abbildung 10: Screenshot aus Unity_Collision_Detection.pdf – Aufbau der Szene (Schober 2019:3).....	21
Abbildung 11: Screenshot aus Unity_Collision_Detection.pdf – Collision Detection (Schober 2019:4).....	22
Abbildung 12: Screenshot aus Unity_Collision_Detection.pdf – Rigidbody (Schober 2019:5).....	22
Abbildung 13: Screenshot aus Unity_Collision_Detection.pdf – Ein Textfeld erzeugen (Schober 2019:6) ..	23
Abbildung 14: Unaufgeräumter Desktop .....	30
Abbildung 15: Aufgeräumter Desktop.....	30
Abbildung 16: Screenshot aus Adobe Captivate 2019: Ein neues Projekt erstellen.....	31
Abbildung 17: Screenshot aus Adobe Captivate 2019: Videodemo Voreinstellungen „Bildschirmbereich“	32
Abbildung 18: Screenshot aus Adobe Captivate 2019: Videodemo Voreinstellungen Bildschirmbereich: „Benutzerdefinierte Größe“ .....	33
Abbildung 19: Screenshot aus Adobe Captivate 2019: Videodemo erweiterte Voreinstellungen .....	34
Abbildung 20: Screenshot aus Adobe Captivate 2019: Screencast bearbeiten .....	35
Abbildung 21: Screenshot aus Adobe Captivate 2019: Voreinstellungen zur Veröffentlichung .....	36
Abbildung 22: Screenshot von GeForce Experience Overlay .....	37
Abbildung 23: Screenshot von GeForce Experience Videoeinstellungen.....	37
Abbildung 24: Screenshot von Screencast mit Captivate: Test des Zoom-Effekts .....	39

Abbildung 25: Screenshot von Adobe Premiere Pro: Ein neues Projekt anlegen .....	41
Abbildung 26: Screenshot von Adobe Premiere Pro: Bildschirmaufnahme in den Media-Browser importieren.....	42
Abbildung 27: Screenshot von Adobe Premiere Pro: Warnung zu nicht übereinstimmenden Clips .....	42
Abbildung 28: Screenshot von Adobe Premiere Pro: Audio entfernen durch „Verknüpfung aufheben“ ....	43
Abbildung 29: Screenshot von Adobe Premiere Pro: Blende-Effekt „Übergang zu Schwarz“ .....	44
Abbildung 30: Screenshot von Adobe Premiere Pro: Exporteinstellungen .....	44
Abbildung 31: Screenshot von Adobe Captivate: Ein neues Projekt anlegen .....	46
Abbildung 32: Screenshot von Adobe Captivate: Designfarben ändern .....	47
Abbildung 33: Screenshot von Adobe Captivate: Designfarbe auswählen .....	48
Abbildung 34: Screenshot von Adobe Captivate: Designfarbe anpassen.....	48
Abbildung 35: Screenshot von Adobe Captivate: Ein Textfeld einfügen .....	49
Abbildung 36: Screenshot von Adobe Captivate: Texteigenschaften über Eigenschaften-Fenster anpassen .....	49
Abbildung 37: Screenshot von Adobe Captivate: Ein Folienvideo einfügen .....	50
Abbildung 38: Screenshot von Adobe Captivate: Anzeigedauer des Textfeldes in der Zeitleiste anpassen (vor der Anpassung).....	51
Abbildung 39: Screenshot von Adobe Captivate: Anzeigedauer des Textfeldes in der Zeitleiste anpassen (nach der Anpassung).....	51
Abbildung 40: Screenshot von E-Learning „1 Unity Einstieg“ auf Folie 17 .....	51
<b>Abbildung 41:</b> Icon linke Maustaste .....	52
<b>Abbildung 42:</b> Icon Mausrad .....	52
<b>Abbildung 43:</b> Icon rechte Maustaste .....	52
Abbildung 44: Screenshot von Adobe Captivate: Neue leere Folie hinzufügen .....	52
Abbildung 45: Screenshot von Adobe Captivate: Veröffentlichungseinstellungen.....	53
Abbildung 46: Screenshot von Adobe Captivate: Projekt veröffentlichen .....	54

## 7. Quellenverzeichnis

### Bücher

Bonfranchi, Thordis (2010): **WordPress - Das Praxisbuch**. Heidelberg : MITP

Chittesh, Jashan (2015): **Das Unity-Buch: 2D- und 3D-Spiele entwickeln mit Unity 5**. Heidelberg : depunkt

Korgel, Daniel (2017): **Virtual Reality-Spiele entwickeln mit Unity®: Grundlagen, Beispielprojekte, Tipps & Tricks**. Dortmund : Carl Hanser

Seifert, Carsten (2017): **Spiele entwickeln mit Unity 5: 2D- und 3D-Games mit Unity und C# für Desktop, Web & Mobile**. München : Carl Hanser

Uhrig, Martin (2012): **Adobe Captivate 6: Erfolgreiche Screencasts und E-Learning-Anwendungen erstellen**. Berlin : epubli

### Vorlesungsunterlagen

Schober, Martin (2019): **Unity Einstieg**. Hochschule Karlsruhe, Fakultät für Informationsmanagement und Medien, Studiengang Kommunikation und Medienmanagement, Skript zur Vorlesung Multimedia, WS2020/21

Schober, Martin (2019): **Unity + C#: Texte eingeben/ausgeben durch Mausclick**. Hochschule Karlsruhe, Fakultät für Informationsmanagement und Medien, Studiengang Kommunikation und Medienmanagement, Skript zur Vorlesung Multimedia, WS2020/21

Schober, Martin (2019): **Unity Collision Detection**. Hochschule Karlsruhe, Fakultät für Informationsmanagement und Medien, Studiengang Kommunikation und Medienmanagement, Skript zur Vorlesung Multimedia, WS2020/21

Schober, Martin (2019): **Unity + C#: Klicks auf 3D-Objekte zählen und ausgeben**. Hochschule Karlsruhe, Fakultät für Informationsmanagement und Medien, Studiengang Kommunikation und Medienmanagement, Skript zur Vorlesung Multimedia, WS2020/21

Schober, Martin (2019): **Unity + C#: Mit Schiebereglern die Größe und die Position eines Objektes verändern**. Hochschule Karlsruhe, Fakultät für Informationsmanagement und Medien, Studiengang Kommunikation und Medienmanagement, Skript zur Vorlesung Multimedia, WS2020/21

Schober, Martin (2019): **Unity + C#: Character Controller**. Hochschule Karlsruhe, Fakultät für Informationsmanagement und Medien, Studiengang Kommunikation und Medienmanagement, Skript zur Vorlesung Multimedia, WS2020/21

Schober, Martin (2019): **Unity + C#: XML-Dateien auslesen und GameObjects zuweisen**. Hochschule Karlsruhe, Fakultät für Informationsmanagement und Medien, Studiengang Kommunikation und Medienmanagement, Skript zur Vorlesung Multimedia, WS2020/21

## Internet

Adobe (2018): „**Erstellen von Videodemos**“

<https://helpx.adobe.com/de/captivate/how-to/create-video-demo.html>

[Stand: 23. August 2018. Zugriff: 20.04.2021, 22:15 MESZ]

E-Learning-Plattformen (o. J.): „**Was ist E-Learning?**“

<https://www.e-learning-plattformen.de/was-ist-e-learning-definition/>

[Stand: 2021. Zugriff: 18.03.2021, 23:35 MESZ]

Ionos (2020): „**Screencast im Überblick: So funktioniert die Bildschirmaufzeichnung**“

<https://www.ionos.de/digitalguide/online-marketing/verkaufen-im-internet/screencast/>

[Stand: 02.09.2018. Zugriff: 03.04.2021, 19:55 MESZ]

Unity Technologies (2020): „**Unity-Manual: Unity User Manual**“

<https://docs.unity3d.com/Manual/index.html>

[Stand: November 2020. Zugriff: 14.11.2020, 18:20 MESZ]

## Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Ausführungen, die anderen veröffentlichten oder nicht veröffentlichten Schriften wörtlich oder sinngemäß entnommen wurden, habe ich kenntlich gemacht.

Die Arbeit hat in gleicher oder ähnlicher Fassung noch keiner anderen Prüfungsbehörde vorgelegen.

---

Ort, Datum

---

Unterschrift