



Fakultät Elektrotechnik Feinwerktechnik
Informationstechnik

DRONEVERSE

Projektbericht im Studiengang Media Engineering

vorgelegt von

Sina Steinmüller, 3637368

Tobias Schwarz, 3627943

Maximilian Richter, 3642353

Lisa Berbig, 3627969

Anusha Kanagarasa, 3631692

Katharina Hölzl, 3579055

Betreuer: Prof. Dr. (USA) Ralph Lano

© 2024

Dieses Werk einschließlich seiner Teile ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

Inhaltsverzeichnis

1	Einführung	2
2	Corporate Identity	4
2.1	Markenidentität	4
2.1.1	USP (Alleinstellungsmerkmal)	4
2.1.2	Zielgruppe	5
2.2	Das Logo	6
2.2.1	Aussage	6
2.2.2	Zusammensetzung	9
2.2.3	Größenangaben	9
2.2.4	Do's / Don'ts	10
2.3	Farbsystem	11
2.3.1	Farbpalette	11
2.3.2	Farbverhältnis	12
2.4	Typografie	13
2.4.1	Schrift	13
2.4.2	Verwendung	13
2.5	Geschäftsausstattung	14
2.5.1	Beispiele	14
2.6	Marketing	17
2.6.1	Marketingstrategie	18
2.6.2	Social Media Strategie	19
2.6.3	Marketingmaterial	24
2.6.4	Zukunftsplan	24
3	Projektbeschreibung	26
3.1	Projektmanagement	28
3.1.1	Obsidian als Notizverwaltungstool	28
3.1.2	Dokumentenablage in Microsoft Teams	30
3.1.3	Terminierte Aufgabenverteilung in Trello	30

3.1.4	Visual Studio Code als kostenfreie Entwicklungsumgebung	31
3.1.5	Versionsverwaltung und Codesharing mit Github	32
3.1.6	Overleaf für die gemeinsame Erstellung des Projektberichts	32
3.2	Interaktive Drohnensteuerung durch Gestenerkennung	33
3.2.1	Technische Ausstattung	34
3.2.2	Gestenerkennungsbasierte Drohnensteuerung in MediaPipe	35
3.2.3	OpenCV und C++-Implementierung in VMware Fusion auf Ubuntu 20.04.5	35
3.2.4	Echtzeit Gestenerkennung mit Tensorflow.js, React.js, Fin- gerpose und Handpose	36
3.2.5	Python mit TensorFlow	42
3.2.6	Resümee und Ausblick	43
3.3	Technische Komponenten der Tello Drohne	45
3.3.1	HD Kamera	45
3.3.2	Barometer	46
3.3.3	Ultraschallsensor	46
3.3.4	Netzwerkchip	46
3.3.5	Status LED	47
3.3.6	Akku	47
3.4	Steuerung der Tello Drohne	48
3.4.1	Bibliothek	48
3.4.2	Gängige Methoden	49
3.4.3	Tastatur-Steuerung	50
3.4.4	Steuerungs-Ablauf	52
3.5	Ansteuerung mehrerer Tello-Drohnen	53
3.5.1	Problembeschreibung	53
3.5.2	Lösungsansatz 1: Zusammenführen der WLAN-Netzwerke	53
3.5.3	Lösungsansatz 2: Promiscuous-Modus	54
3.5.4	Lösungsansatz 3: ESP32	54
3.5.5	Lösungsansatz 4: Drohnen in ein externes Netzwerk aufnehmen	55
3.5.6	Fazit zur Steuerung mehrere Drohnen	56
3.6	Kollisionsverhinderung	57
3.6.1	Kollisionsverhinderung mit der Umwelt	57
3.6.2	Kollisionsverhinderung der Drohnen untereinander	57
3.7	Externe Komponenten	62
3.7.1	Heliumluftballons für die Tello-Drohne / Quadcopter	62

3.7.2	Sicherheitsnetz	63
3.7.3	LED Module	63
3.8	Quadcopter	64
3.8.1	Cheerson CX-100	64
3.8.2	Erste Versuche	65
3.8.3	nRF24-Transceiver	66
3.8.4	Blimp	70
3.8.5	Aktueller Stand	73
3.8.6	Ziele für nächstes Semester (SoSe24/25)	73
4	Stand der Technik	75
4.1	Gestenerkennung	75
4.2	Drohnensteuerung	76
4.2.1	Drohne (allgemein)	76
4.2.2	Drohnenschwarm	77
5	Ausblick	79
5.1	Interaktive Drohnensteuerung	80
5.1.1	Hindernisparcour	80
5.1.2	Langzeitbelichtung	80
5.1.3	Drohnenmultiplayergame	80
5.2	Raumgestaltung	82
5.2.1	Umsetzung und Durchführung der Präsentation	83
6	Zusammenfassung	85
	Abbildungsverzeichnis	86
	Listings	88
	Bibliographie	89

Kapitel 1

Einführung

(Anusha Kanagarasa)

Im Rahmen des fünften und sechsten Semesters im Studiengang Media Engineering an der Technischen Hochschule Nürnberg findet eine Projektarbeit statt. Besonders hervorzuheben ist, dass dieses langfristige Projekt im Studiengang BME ein Schwerpunkt ist. Dabei werden den Studierenden verschiedene Projekte angeboten, aus denen sie wählen können. Nach einer anschließenden Abstimmung werden sie Gruppen zugeteilt. Dieses Semester bietet den Studierenden die wertvolle Möglichkeit, ihr theoretisch erlerntes Wissen in die Praxis umzusetzen.

Unser Projekt behandelt die Forschung und Entwicklung einer interaktiven, auf Gesten basierenden Drohnensteuerung. Unterstützt werden die Drohnen dabei von mit Helium gefüllten Luftballons, welche in der Aufgabenstellung vorgegeben waren.

Die Luftballons erfüllen hier eine entscheidende Rolle: Sie dienen nicht nur als ästhetisches Element, sondern haben auch einen funktionalen Zweck. Durch die geschickte Platzierung der Luftballons wird die Schwebefähigkeit der Drohne erhöht, wodurch sie länger in der Luft bleiben kann. Gleichzeitig wird die Empfindlichkeit der Drohne gegenüber Steuerungseingaben verringert, was eine präzisere und sanftere Kontrolle ermöglicht.

Unser Ziel ist es, dieses innovative Steuerungskonzept als interaktive Kunstinstallation zu präsentieren. Wir planen es nicht nur auf dem Projekttag unserer Hochschule, sondern auch auf öffentlichen Veranstaltungen wie der Blauen Nacht zu zeigen, sofern wir am Projekttag eine positive Resonanz erhalten. Die Blaue Nacht ist ein jährliches Kulturevent, bei dem verschiedene künstlerische Darbietungen und Installationen in der Stadt präsentiert werden.

Insgesamt streben wir danach, nicht nur eine technische Innovation zu entwickeln, sondern ein ganzheitliches Erlebnis zu schaffen, welches die Grenze zwischen Kunst, Technologie und Interaktion verschwimmen lässt und die Vorstellungskraft der Betrachter anregt. Unsere interaktive Drohnensteuerung mit Luftballons soll nicht nur beeindrucken, sondern auch inspirieren.

Mit diesem Ziel vor Augen haben wir uns an die Arbeit gemacht, um DRONEVERSE von einer Idee zu einer erlebbaren Realität zu machen. In diesem Bericht werden wir den gesamten Prozess der Konzeption, Umsetzung und Durchführung dieses aufregenden Projekts beleuchten.

Kapitel 2

Corporate Identity

2.1 Markenidentität

2.1.1 USP (Alleinstellungsmerkmal)

(Anusha Kanagarasa)

DRONEVERSE stellt nicht nur ein fiktives Unternehmen im Bereich der Drohnentechnologie dar. Unser Anspruch geht über das bloße Erbringen guter Arbeit hinaus. Der Fokus wird auf Innovation und Kreativität gesetzt, um ein unvergleichliches Erlebnis zu schaffen, welches die Besucher begeistern und faszinieren soll. Unser Alleinstellungsmerkmal liegt in der innovativen Nutzung von Luftballons zur Unterstützung der Drohnensteuerung. Diese Verwendung verfolgt gleichzeitig eine künstlerische Ästhetik. Zusätzlich ermöglicht die Gestensteuerung eine direkte Interaktion mit der Drohne. Diese einzigartige Kombination hebt uns von anderen Mitbewerbern ab und macht unser Projekt besonders.

Die Vorstellung der Ergebnisse wird in einer stimmigen und harmonischen Gesamterscheinung präsentiert. Dies beginnt bei dem Onlineauftritt bis hin zu allen Printmedien. Großer Wert wird darauf gelegt, dass jeder Aspekt unserer Marke hohe Standards einhält. Potenzielle Interessenten sollen sofort erkennen, dass Projekte, die von uns durchgeführt werden, nicht nur „sehr hochwertig“ sind, sondern darauf abzielen, ihre Wünsche in vollem Umfang zu erfüllen.

Insgesamt wird nicht nur eine technische Innovation präsentiert, sondern ein ganzheitliches Erlebnis geschaffen, das die Grenze zwischen Kunst, Technologie und

Interaktion verschwimmen lässt. DRONEVERSE ist nicht einfach nur ein Unternehmen, in dem Drohnen gesteuert werden. Vielmehr ist das Ziel, den Besuchern der Präsentation ein unvergessliches Erlebnis zu liefern.

Wir differenzieren uns im Wettbewerb mit der Konkurrenz durch die folgenden Merkmale:

- Seriosität und Eleganz
- interaktive Kunstinstallation
- Fokus auf Benutzerfreundlichkeit
- Multifunktionale Einsatzmöglichkeiten

2.1.2 Zielgruppe

1. Unternehmen: DRONEVERSE bietet Unternehmen eine einzigartige Möglichkeit, ihre Produkte und Dienstleistungen auf innovative und beeindruckende Art und Weise zu präsentieren. Es kann als Teil der Marketing- oder Branding-Strategie eingesetzt werden, um eine starke visuelle Wirkung zu erzielen und das Interesse von Zielgruppen zu wecken.

2. Marketingagenturen: Marketingagenturen können DRONEVERSE nutzen, um ihren Kunden kreative und neuartige Werbekampagnen anzubieten. Die Kombination aus Drohnentechnologie und Kunstinstallation kann dazu beitragen, sich von der Masse abzuheben und das Publikum zu beeindrucken. Zusätzlich können Luftballons kostengünstig an die Drohne angebracht werden, um weitere visuelle Effekte und Aufmerksamkeit für die Kampagne zu generieren.

3. Kunstinteressierte jeden Alters: DRONEVERSE wird auch Kunstliebhaber ansprechen, die sich sowohl für moderne Technologie als auch für künstlerische Konzepte interessieren. Die Verknüpfung von Technologie und Kunst, einschließlich des Einsatzes von Luftballons, soll das Publikum begeistern und ihre künstlerische Neugierde wecken.

4. Bildungseinrichtungen: Schulen, Universitäten und Bildungseinrichtungen können DRONEVERSE nutzen, um Schülern und Studierenden Technologie und kreative Konzeptionen näherzubringen. Das Projekt kann dazu beitragen, das

Interesse an STEM-Fächern (Science, Technology, Engineering and Mathematics) und Kunst zu wecken und Schülern neue Lernmöglichkeiten zu bieten.

5. Technologie-Enthusiasten: Menschen, die ein Interesse an Technologien wie Drohnen und interaktiven Installationen haben, sollen von DRONEVERSE fasziniert werden. Die Möglichkeit, diese Technologien in Aktion zu erleben und auszuprobieren, soll ihr Interesse wecken.

Durch die Vielseitigkeit und visuelle Attraktivität von DRONEVERSE soll das Projekt eine breite Palette von Zielgruppen ansprechen und begeistern.

2.2 Das Logo

2.2.1 Aussage

Name: DRONEVERSE

Logo: modernes, minimalistisches Logo, das Technologie und Kunst symbolisiert.

Farbschema: kühle Farbtöne wie Blau- und Lilatöne, um die Atmosphäre einer mit Luftballons ausgeschmückten Höhle zu betonen.

Das Logo dient als zentrales Element, um unsere Marke an verschiedenen Berührungspunkten erkennbar zu machen. Es soll nicht nur einprägsam und wiedererkennbar sein, sondern auch die Aufmerksamkeit unserer Zielgruppe auf sich ziehen.

Während des Prozesses haben wir darüber nachgedacht, wie wir unser Projekt präzise präsentieren können. Dabei wurden verschiedene Inspirationen in Betracht gezogen, angefangen von Drohnenskizzen bis hin zu fantasievollen Zeichnungen von Drachen und Tierarten, die Eigenschaften mit unseren Drohnen teilen.

Hier sind einige Skizzen, die während des Designprozesses entstanden sind. Jede Skizze repräsentiert einen Schritt in unserem kreativen Prozess und zeigt die Vielfalt der Ideen, die wir in Betracht gezogen haben:

Nach langen Überlegungen und zahlreichen Experimenten mit Kombinationen aus Drohne und Luftballon haben wir uns schließlich für ein Wappentier entschieden - den Kolibri. Diese Entscheidung basiert auf den sehr ähnlichen Fähigkeiten des Kolibris im Vergleich zu Drohnen.

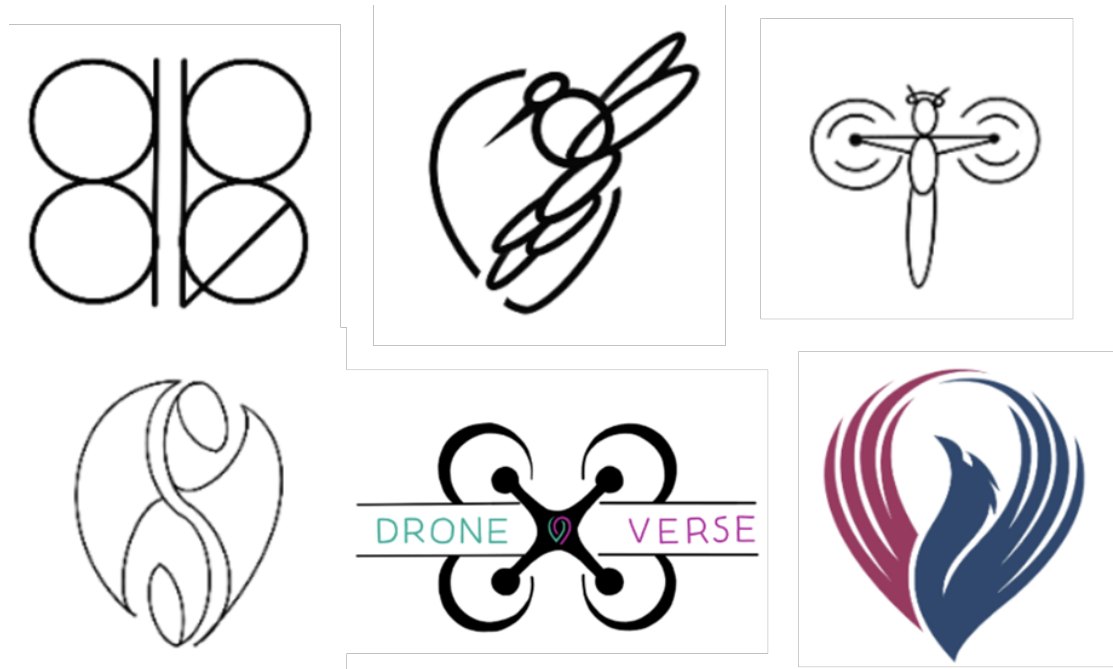


Abbildung 2.1: Unsere Logo-Skizzen

Motivation hinter dem Kolibri als Wappentier:

- Ähnliche Flugeigenschaften: Der Kolibri symbolisiert die fliegerischen Fähigkeiten, welche auch unsere Drohne beherrscht. Der Kolibri ist in der Lage, in der Luft zu schweben und blitzschnell durch die Umgebung zu manövrieren. Dies spiegelt die Agilität und das schnelle, präzise Flugverhalten unserer Drohnen wider.
- Eleganz und Anmut: Der Kolibri verleiht der Wort-Bild-Marke eine natürliche Eleganz und Anmut. Diese Eigenschaften unterstreichen die ästhetische Dimension unseres Projekts und heben es von rein technischen Darstellungen ab.

Die Wahl des Kolibris als Wappentier verleiht der Marke nicht nur eine visuelle Identität, sondern verknüpft auch die fliegerischen Eigenschaften dieses faszinierenden Vogels mit den einzigartigen Merkmalen unserer Drohnenshow und Kunstinstallation.



DRONEVERSE

Abbildung 2.2: DRONEVERSE-Logo

2.2.2 Zusammensetzung



DRONEVERSE

Darf nur allein (ohne Wortmarke) stehen, wenn DRONEVERSE anderweitig auf der Seite genannt ist.

Darf nur nach Absprache mit dem Brand Management alleine stehen.

2.2.3 Größenangaben

Das Logo darf nicht kleiner als 20mm verwendet werden.

Empfohlene Logohöhe bei verschiedenen DIN-Formaten:

- DIN A5 = Höhe ca. 35 mm
- DIN A4 = Höhe ca. 50 mm
- DIN A3 = Höhe ca. 80 mm
- DIN A2 = Höhe ca. 120 mm usw.

2.2.4 Do's / Don'ts



Das Logo darf nicht verzerrt werden.



Das Logo darf nicht gedreht werden



Die Wortmarke darf nicht willkürlich vom Logo getrennt werden.



Das Logo darf nicht angeschnitten werden.



DRONEVERSE

Die Bildmarke darf nur in Ausnahmefällen weggelassen werden.

2.3 Farbsystem

2.3.1 Farbpalette

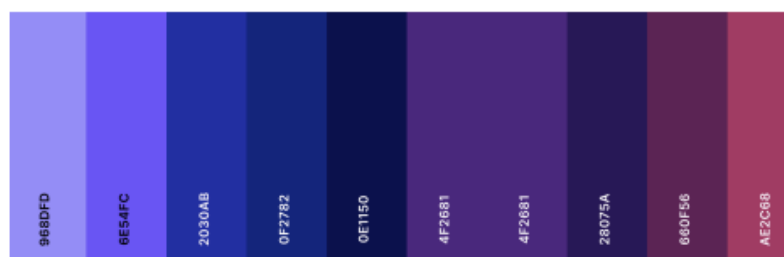
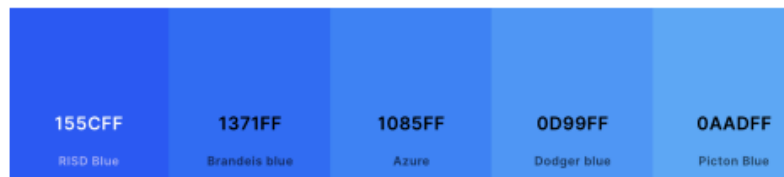
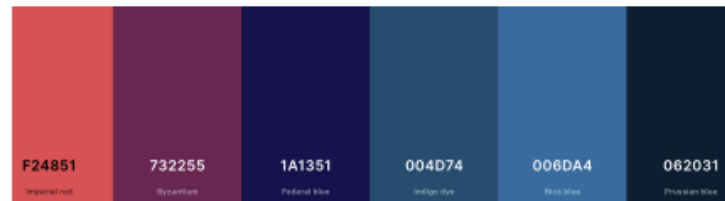


Abbildung 2.3: Corporate Design Farben

2.3.2 Farbverhältnis

Die Farbpalette von DRONEVERSE wurde sorgfältig ausgewählt, um eine einladende und gleichzeitig futuristische Atmosphäre zu schaffen. Dominiert von kühlem Blau- und Violetttönen, erzeugen die Farben eine Aura von Technologie und Innovation, die optimal mit dem Thema des Projekts harmoniert. Diese Farbtöne symbolisieren die faszinierende Welt der Drohnenshows und Kunstinstallationen, in der sich unsere Besucher auf eine Reise voller Entdeckungen begeben.

Durch die Ergänzung von tieferen Nuancen entsteht ein visueller Kontrast, der dem Farbschema Tiefe und mehrere Dimensionen verleiht. Diese tiefen Töne dienen dazu, die visuelle Wirkung zu verstärken, während sie gleichzeitig eine elegante und ansprechende Ästhetik bewahren.

Die lebendigen Blautöne in der Palette verleihen dem Gesamtbild eine erfrischende Note. Sie bringen Leben und Bewegung in das Projekt und erwecken die Drohnenshow zum Leben. Diese Farben repräsentieren die kreative und inspirierende Atmosphäre von DRONEVERSE.

Insgesamt schafft die geschickte Kombination dieser Farbtöne eine harmonische und visuell ansprechende Umgebung, die das Projekt DRONEVERSE in Szene setzt und das Publikum in eine faszinierende Welt voller Technologie, Kunst und Unterhaltung entführt.

2.4 Typografie

2.4.1 Schrift



Abbildung 2.4: Schriftart

2.4.2 Verwendung

Für eine klare und verständliche Kommunikation verwenden wir eine serifenlose Schriftart mit klar strukturierten Paragraphen. Dies gilt sowohl für die Plakate als auch für die Social-Media-Inhalte, um eine breite Zugänglichkeit sicherzustellen.
für Logo, Plakat, Flyer:

- Pragmatica Extended Bold

für Website, Social Media, Fließtext:

- Avenir Light (Fließtext)
- Avenir Medium (Überschriften)
- Avenir Heavy (Hervorhebung wichtiger Informationen)

2.5 Geschäftsausstattung

2.5.1 Beispiele



Abbildung 2.5: Mockup-Notizbuch



Abbildung 2.6: Mockup-T-shirt



Abbildung 2.7: FlyerFront



Abbildung 2.8: FlyerBack

2.6 Marketing

(Katharina Hölzl)

Die Marketingstrategie von DRONEVERSE konzentriert sich vor allem darauf, uns durch außergewöhnliche Ideen, kreative Ansätze und den Einsatz innovativer Technologien klar von anderen Projekten abzuheben. Das Ziel ist es, Ballons und Drohnen in einer harmonischen Flugshow zu vereinen und DRONEVERSE zu einem unvergesslichen Erlebnis zu machen. Unsere Show ist einzigartig, da das Konzept nach umfangreicher Recherche bisher unverwechselbar bleibt. Während andere Veranstalter entweder nur Drohnen für ihre Shows nutzen oder Ballons als statische Kunstinstallationen einsetzen, erschaffen wir eine dynamische Show.

So gibt es zum Beispiel die Firma Lunatx SFX, welche sich auf Special Effects und Eventinszenierungen spezialisiert hat und dabei unter anderem auch Drohnen zum Einsatz bringt. [31]

Die Firma SKYMAGIC ist laut eigener Aussage, „the world’s leading drone light show company, based in the United Kingdom, Singapore, and the USA. We are an award-winning team of creatives, technologists and storytellers“. [44] Seit 2015 begeistern sie mit ihren hochprofessionellen und komplexen Drohnenshows die Zuschauer.

Während der Recherche nach statischen Ballonshows stach die Firma Ballonetti heraus, welche sich auf Kunstinstallationen, Bühnengestaltung und Fotobackdrops spezialisiert hat. [5]

Vom 21 März - 1 Mai 2022 gab es eine Ausstellung mit dem Namen „Let’s Fly“ im Ballon Museum Rom. „Dieses fantasievolle Museum wurde berühmt für seine aufblasbaren Kreationen, darunter riesige Kunstwerke und Pools aus bunten Ballons, die Erwachsene und Kinder gleichermaßen begeistern.“ [4]

2.6.1 Marketingstrategie

Die Marketingstrategie bis zur Abschlusspräsentation konzentriert sich hauptsächlich auf Social Media. Um herauszufinden, welche Plattform am geeignetsten ist, wurden mehrere unter die Lupe genommen. Dafür wurde unter anderem die Webseite Hootsuite verwendet. Hootsuite ist ein Unternehmen, das sich auf Social-Media-Management-Systeme spezialisiert hat. Die Plattform ermöglicht es Einzelpersonen, Unternehmen und Organisationen, ihre Aktivitäten in verschiedenen sozialen Netzwerken zu verwalten, zu planen und zu analysieren. Mit Hootsuite können Benutzer Beiträge erstellen, Zeitpläne festlegen, verschiedene soziale Netzwerke gleichzeitig überwachen, Nachrichten verfolgen und Analysen über die Leistung ihrer Inhalte erhalten. Die Plattform ist besonders nützlich für Marketing- und Kommunikationsteams, die ihre Präsenz in sozialen Medien optimieren und effektiv verwalten möchten. [22]

Basierend auf eigenen Einschätzungen und Informationen der Webseite ergibt sich folgende Schlussfolgerung:

Snapchat wird als Social Media Plattform aufgrund der nur 24 Stunden sichtbaren Stories ausgeschlossen.

X (ehemals Twitter) hat aufgrund seiner schlechten Reputation und des unzureichenden Supports für uns an Attraktivität verloren. Außerdem erfüllt diese Plattform nicht unsere Anforderung, möglichst unterschiedliche Arten von Beiträgen posten zu können.

Obwohl TikTok die Möglichkeit bietet, kurze Videos zu erstellen, um sehr schnell große Aufmerksamkeit zu erlangen, fehlt es an der Vielseitigkeit der veröffentlichten Inhalte. Auch die begrenzte Wortanzahl, die in der Kommentarspalte zur Verfügung steht, macht TikTok unattraktiv.

Die Wahl fiel daher auf Instagram, eine Plattform, die von Millionen Menschen aller Altersgruppen genutzt wird und verschiedene Optionen bietet, das Projekt bekannt zu machen und Reichweite zu generieren.

Es bleibt jedoch noch offen, ob wir den Instagram-Account mit Facebook verknüpfen werden. Da beide Plattformen zu META gehören, besteht die Möglichkeit, die Konten zu verbinden, sodass Stories oder Beiträge, die auf einer Plattform gepostet werden, automatisch auch auf der verknüpften Plattform erscheinen.

2.6.2 Social Media Strategie

Um eine professionellere Projekt-Präsenz zu gewährleisten, wurde ein geschäftliches Konto erstellt und als „Content Creator“ eingetragen. Das ermöglicht Einblicke in die von Instagram bereitgestellten Statistiken. Zum Beispiel können wir sehen, ob Follower unsere Beiträge teilen oder auf andere Art und Weise mit ihnen interagieren. Stand 20.02.2024 verzeichnen wir insgesamt 33 Follower auf Instagram und haben 9 Beiträge veröffentlicht.

Die Produktion von Content im Voraus ist wichtig, um ihn zur optimalen Zeit veröffentlichen zu können. Unser Ziel ist es, eine möglichst breite Masse an Nutzern zu erreichen. Daher ist geplant regelmäßig Beiträge in bestimmten Zeitabständen und zu geeigneten Uhrzeiten zu veröffentlichen. Der Instagram-Algorithmus bevorzugt vor allem Aktualität, das bedeutet, man postet dann, wenn möglichst viele Menschen auf Social Media aktiv sind. Hierzu wurde auf Hootsuite recherchiert, wann genau die perfekte Uhrzeit ist um möglichst viele Menschen zu erreichen. [28] Hootsuite hat mehr als 30.000 Beiträge von Unternehmen verschiedenster Größenordnung analysiert und kam zu dem Ergebnis: Mittwoch, 11 Uhr Pazifische Standard Zeit ist die beste Zeit zum Posten, das heißt um 20 Uhr unserer Zeit. Des Weiteren ist es ratsam, dann zu Posten, wenn die User unterwegs sind, zum Beispiel auf dem Weg zur Arbeit zwischen 6 und 8 Uhr oder auf dem Heimweg zwischen 16 und 20 Uhr, was sich auch mit den Ergebnissen der Recherche deckt.

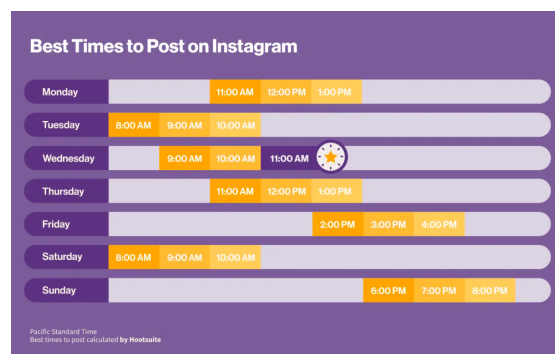


Abbildung 2.9: Statistik zu Instagram

Die Aktivität mit täglichen Story-Beiträgen ist ein weiterer Schwerpunkt, damit die Follower kontinuierlich mit neuen Inhalten versorgt werden und DRONEVERSE immer im Gedächtnis der Interessierten bleibt.

Auf unserem DRONEVERSE-Account verfolgen wir aktiv andere Creator im Bereich der Drohnensteuerung und Kunstinstallationen sowie Accounts der Technischen Hochschule. Unter diesen Accounts wird fleißig kommentiert, Stories und Beiträge werden geliked, der Content wird geteilt und interessante Beiträge werden gespeichert. Durch diese Interaktionen möchten wir den Instagram-Algorithmus nutzen und unsere Präsenz in der Community stärken. Durch die tägliche Benutzung von Instagram und regelmäßige Analyse der Statistiken, versuchen wir die Bedürfnisse und Interessen unserer Follower zu verstehen und die Algorithmen zu unseren Gunsten zu verwenden.

Für die Beiträge werden stets hochwertige Aufnahmen verwendet und mit projektrelevanten Hashtags versehen, denen bereits Millionen von Nutzern folgen. Dadurch erhöhen sich die Chancen, dass sie von gleichgesinnten Kunst- und Technikliebhabern gesehen werden und weitere Follower dazu gewonnen werden.

In Zukunft ist geplant, verstärkt mit bereits etablierten Accounts zu interagieren und Kooperationen einzugehen, um deren Reichweite für uns zu nutzen. Für jeden Post wird außerdem unser eigener Marken-Hashtag DRONEVERSE verwendet.

Alle Beiträge entsprechen dem Corporate Identity Schema, das bedeutet, dass sich an die Unternehmensfarben und Schriftarten gehalten wird und nicht in vielen verschiedenen Farben gepostet wird.

Außerdem werden die Apps „My Feed“ und „Mojo“ verwendet, um den produzierten Content zu bearbeiten und in bereits vorgefertigte Grids zu legen, was es ermöglicht, den Feed zu sehen und nach eigenen Wünschen zu bearbeiten. Mit My Feed hat man einen sehr guten Überblick über das Instagram-Konto, was das Planen der nächsten Beiträge erleichtert. Durch Mojo können Beiträge optisch und qualitativ verbessert werden, da man Audiospuren über Videos und Beiträge legen kann, eine große Auswahl an Schriftarten und einen Videoeditor hat, mit dem Beiträge schnell und einfach bearbeitet werden können.

Die wichtigsten Beiträge und Informationen werden in den Highlights hinterlegt, zum Beispiel Stories und Beiträge, die auf die Projektpräsentation aufmerksam machen oder den Projektfortschritt zeigen.

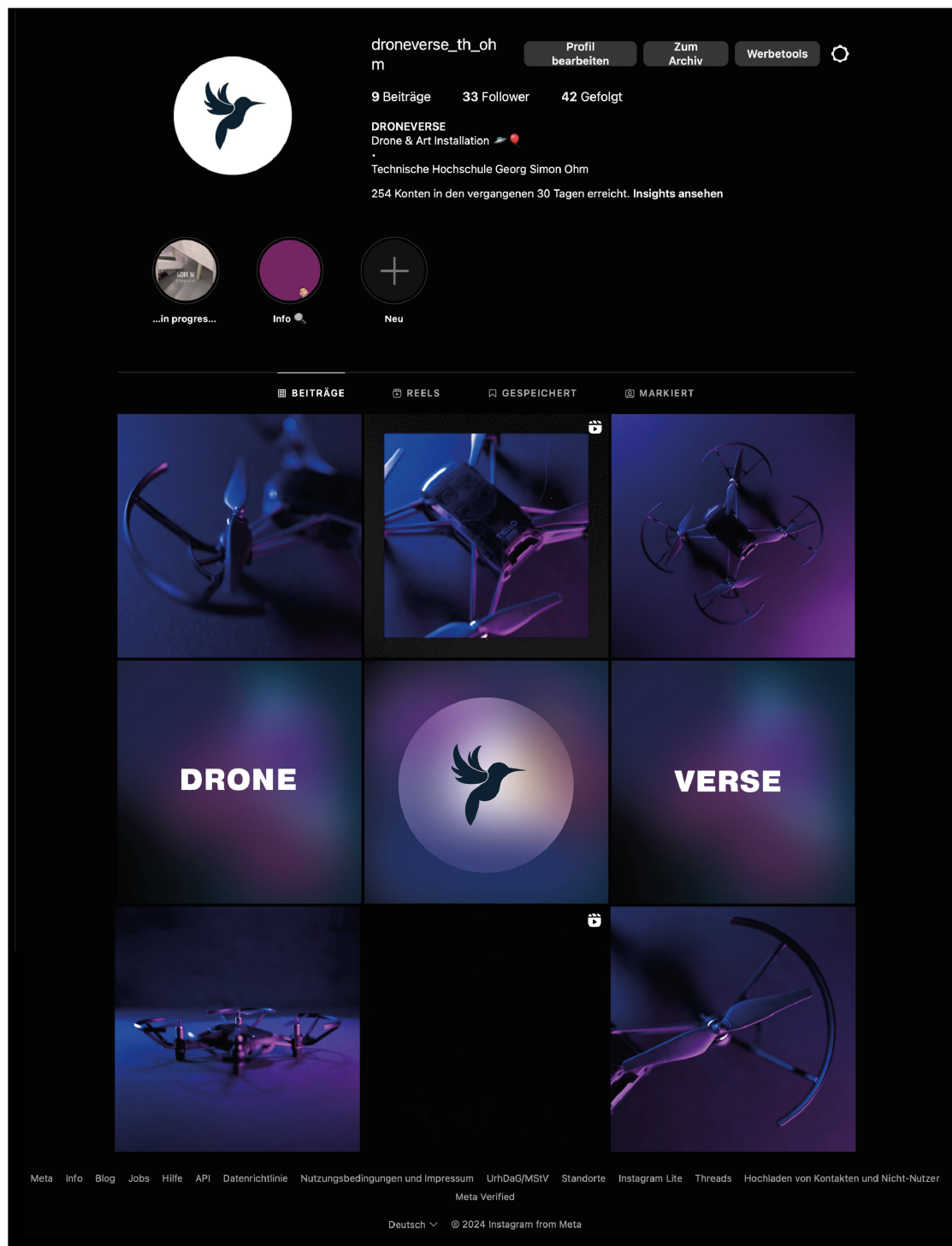


Abbildung 2.10: Instagram DRONEVERSE

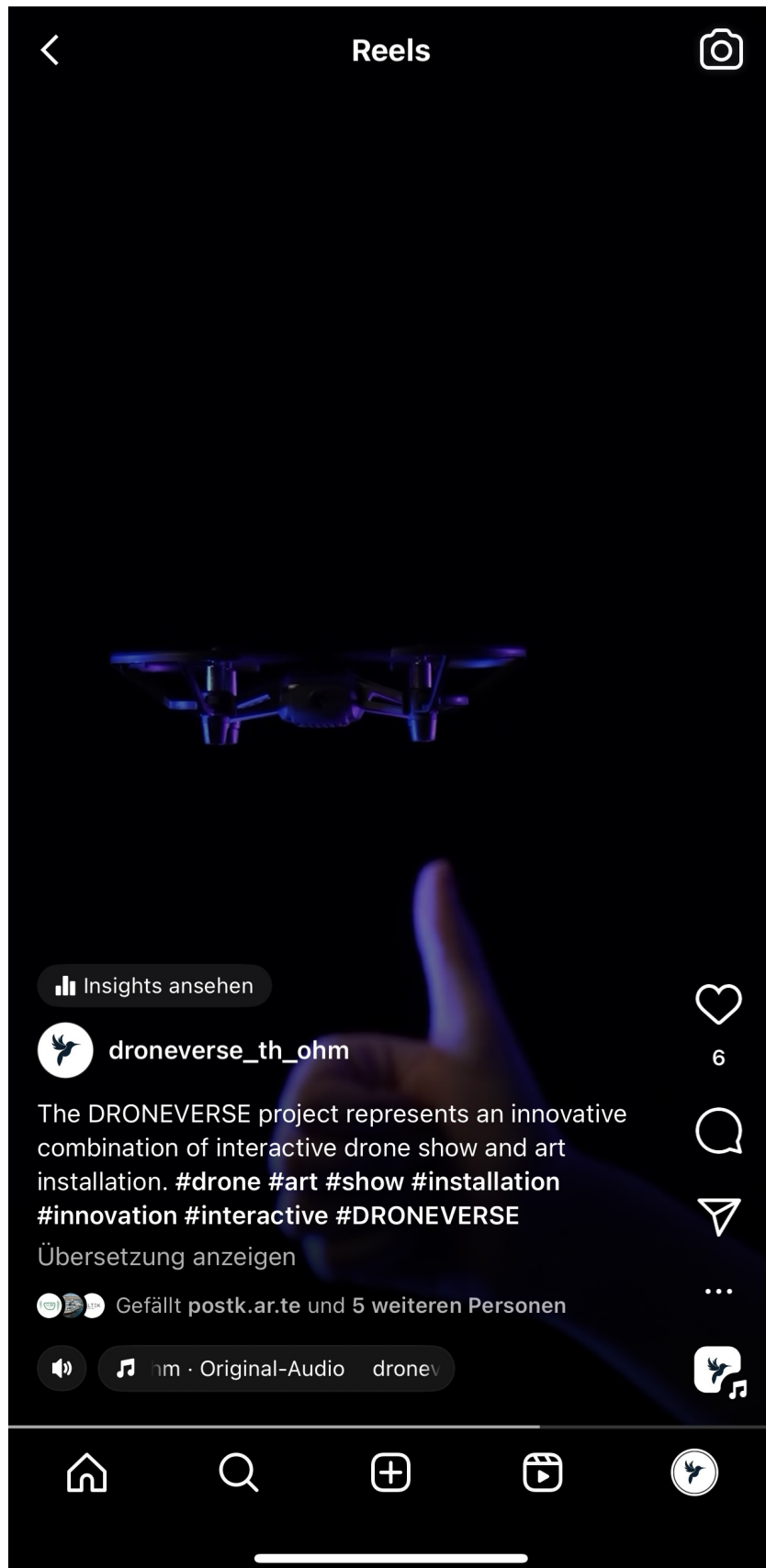


Abbildung 2.11: Beitrag DRONEVERSE

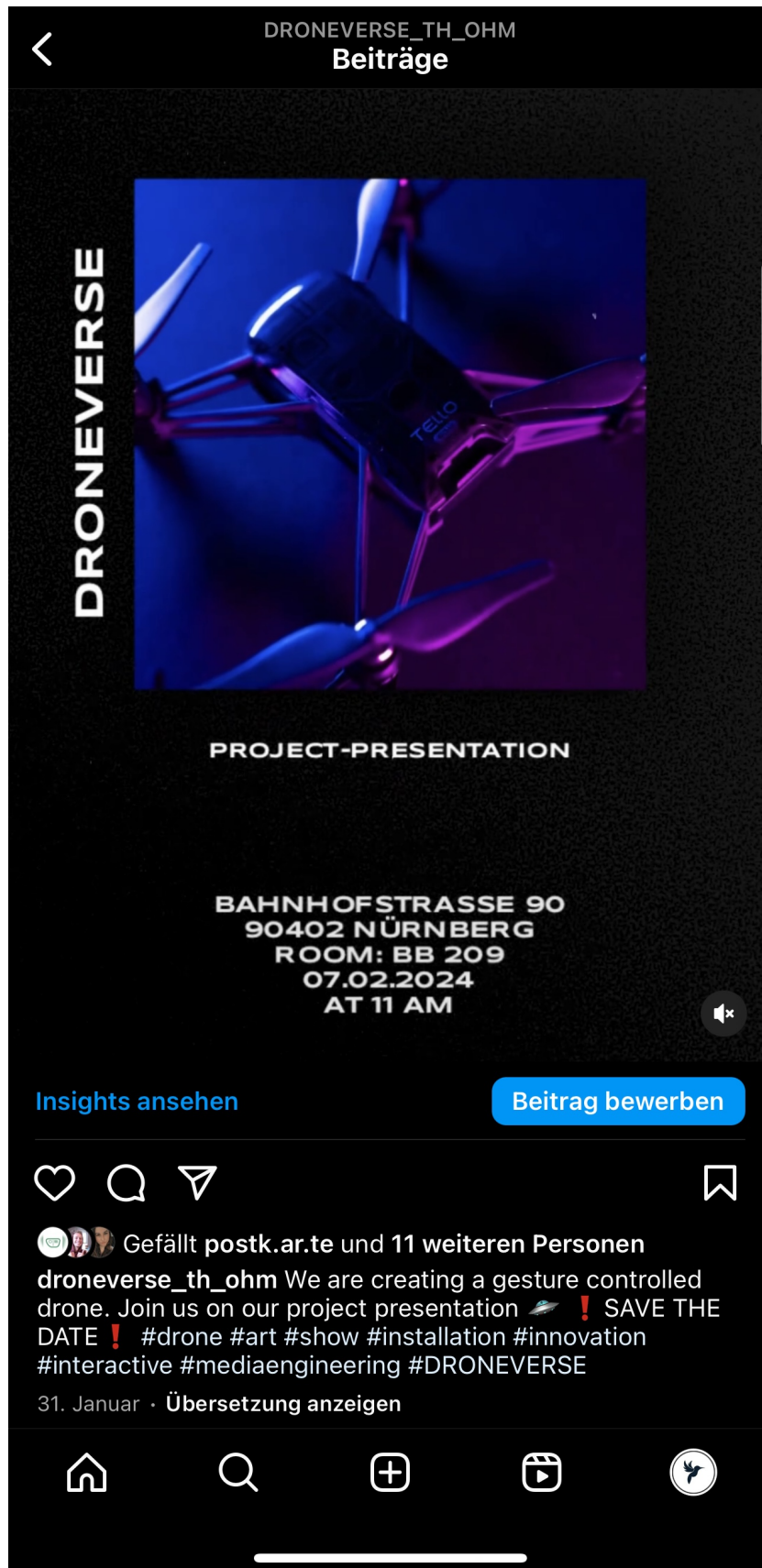


Abbildung 2.12: Beitrag DRONEVERSE

2.6.3 Marketingmaterial

Das Marketing-Material umfasst unseren Instagram-Kanal, der das Content-Marketing repräsentiert. Bislang wurden Plakate erstellt, um auf unser Projekt für die Zwischenpräsentation aufmerksam zu machen. Darüber hinaus haben wir mehrere Instagram-Stories und Beiträge veröffentlicht.

2.6.4 Zukunftsplan

In Zukunft ist geplant, mich mit Blender vertraut zu machen, um einen Trailer für unser Projekt zu erstellen. Meine Idee ist es, in Blender einen Kolibri zu modellieren, der durch Landschaften aus Wäldern und Flusstälern fliegt. Dabei hinterlässt er eine Glitzerspura, um dem Ganzen eine magische und mysteriöse Atmosphäre zu verleihen. Der Kolibri wird von weiteren Artgenossen begleitet und gemeinsam finden sie den Weg in unsere Ballonhöhle, wo sie von glitzernden Lichtern, Ballons und zahlreichen anderen Kolibris empfangen werden. Die Kolibris sind unser Wappentier und sollen symbolisch für unsere Drohnen stehen. Der entstandene Film wird eine Art Werbefilm sein und soll auch am Tag der Abschlusspräsentation gezeigt werden.

Zusätzlich ist geplant, mithilfe von Blender einen detaillierten Raumplan für unser Projekt zu erstellen, sobald wir mit absoluter Sicherheit wissen, welcher Raum uns für unsere abschließende Präsentation zugewiesen wird.

Blender ist eine freie Software, die eine breite Palette von Werkzeugen für die Medienproduktion bietet. Die wichtigsten Features von Blender umfassen eine vollintegrierte 3D-Content-Creation-Suite, die eine Vielzahl von Werkzeugen wie Modellierung, Rendern, Animation und Rigging, Videoschnitt, VFX, Komposition, Texturierung und verschiedene Simulationen bietet. Blender ist plattformübergreifend einsetzbar und verfügt über eine OpenGL-Benutzeroberfläche, die auf allen gängigen Plattformen einheitlich ist und mit Python-Skripten anpassbar ist. Durch die hochwertige 3D-Architektur ermöglicht Blender einen schnellen und effizienten Arbeitsablauf. [17]

Des Weiteren haben wir vor, neue Poster zu entwerfen und auf dem Gelände der Technischen Hochschule aufzuhängen. Zusätzlich wollen wir Flyer und eine Broschüre entwickeln, welche das Projekt, seine Idee, Ziele und die gesamte Umsetzung

kurz und verständlich erklären. Für die Abschlusspräsentation ist es auch unser Ziel, eine kurze Pressemitteilung zu verfassen.

Darüber hinaus überlegen wir, Visitenkarten anzufertigen, auf denen der Name der Webseite und unser Instagram-Kanal verlinkt sind. Kurz vor unserer finalen Veranstaltung planen wir, kleine Sticker mit einem QR-Code im gesamten Hochschulgelände zu verteilen, um die Aufmerksamkeit auf unser Projekt zu lenken. Außerdem möchten wir als kleine Belohnung für unsere Besucher weitere Sammelsticker mit unserem Logo, dem Kolibri oder Sticker mit kleinen Drohnen und Luftballons anbieten.

Eine kleine Herausforderung besteht darin, nicht nur das Instagram-Konto konsequent zu führen, sondern auch eine eigene Webseite für DRONEVERSE zu gestalten. Diese soll ein ansprechendes und modernes Design haben und eine breite Zielgruppe ansprechen.

Kapitel 3

Projektbeschreibung

(Sina Steinmüller)

DRONEVERSE soll die Frage beantworten, ob man aus zwei DJI Tello (Kapitel 3.3) Entwicklungsdrohnen, zwei Quadkoptern (Kapitel 3.8) und einigen mit Helium gefüllten Luftballons (Kapitel 3.7) ein sehenswertes und interaktives Media Engineering Projekt realisieren kann.

Abgesehen vom Titel „interaktive Drohnen mit Luftballons“ wurde bei der Projektarbeit die Umsetzung aller Teilgebiete freigestellt und der erste Teil des Projekts, den dieser Zwischenbericht beschreibt, diente hauptsächlich der Ideenfindung für ein Konzept und der Umsetzbarkeit dieses Konzepts. Im nun folgenden zweiten Halbjahr wird das Projekt finalisiert, sodass es bei der Abschlusspräsentation Ende Juli 2024 vorzeigbar ist.

Alle Projekte haben in etwa ein Budget von 100€ für die Realisierung zur Verfügung. Darüber hinaus darf sich jede Gruppe Material aus der Hochschule ausleihen, wie beispielsweise 3D Drucker, Drohnen oder verschiedene Kameras. Ein gelungenes Proof of Concept während der Zwischenpräsentationen bietet jedoch die Möglichkeit einer Erhöhung dieses Budgets, sofern von diesen Ausgaben Studenten eines höheren Semesters ebenfalls profitieren können.

Das DRONEVERSE Team besteht aus sechs Media Engineering Studenten des 5. Semesters im Alter von 20 bis 27 Jahren, die sich durch die Onlineprojektwahl zufällig zusammengefunden haben. Neben der inhaltlichen Arbeit am Projekt geht es während dieses Jahres auch um die Fähigkeiten zur Selbstorganisation dieses zusammengewürfelten Teams von unerfahrenen Studenten. Hierbei insbesondere um die Stärkung der individuellen Persönlichkeit sowie der Herausforderung, selbstständig geeignete Aufgaben für das Projekt zu finden und umsetzen zu können,

ohne von einem Chef konkrete Anweisungen zu erhalten. Alternativ besteht auch die Möglichkeit, ein konkreter Teamleiter des Projekts zu werden, der entsprechende Anweisungen vergibt, sofern die Selbstständigkeit scheitert. Das Projektjahr gibt jedem Studenten die Chance, Einblicke in unbekannte Themengebiete zu erhalten, vorhandene Stärken ins Team einzubringen und risikofrei, ohne Angst vor Fehlern oder Scheitern, eigene Schwächen zu entdecken und mit ihnen umzugehen.

Eine weitere Herausforderung wie auch Chance, ist die besondere Freiheit in der Wahl der verwendeten Mittel und Methoden und der nach oben offenen Skalierbarkeit des Projekts. Während im Arbeitsleben einige Softwaretools und Herangehensweisen bereits vordefiniert sind, ist es bei einem Projekt dieser Art gänzlich freigestellt, welche und wie viele Tools man verwenden möchte. Diese Freiheit ist Chance und Herausforderung zugleich. Je nach Motivation und Können der einzelnen Teammitglieder sowie des Teams als Ganzes ist daher sowohl eine rudimentäre, gestenbasierte Drohnensteuerung möglich, als auch beispielsweise ein als Kunstinstallation auf öffentlichen Veranstaltungen verwendbares Erlebnis. Wie dieses Erlebnis am Ende aussehen könnte, wird im Ausblick (Kapitel 5) skizziert.

Die folgende Projektbeschreibung liefert einen Überblick über die erste Hälfte des Projektjahres, chronologisch beginnend mit der größten Hürde, dem Projektmanagement (Kapitel 3.1). Anschließend werden die einzelnen Projektteile unabhängig voneinander detailliert beschrieben: die Gestenerkennung (Kapitel 3.2), die technischen Komponenten (Kapitel 3.3), die Steuerung der Tello Drohnen (Kapitel 3.4), Externe Komponenten (Kapitel 3.7) sowie die Proof of Concept Findung der Quadcopter (Kapitel 3.8).

3.1 Projektmanagement

(Sina Steinmüller)

Der Einstieg in die DRONEVERSE Projektarbeit begann zunächst einmal mit dem Kennenlernen der einzelnen Teammitglieder untereinander und dem Brainstorming für das kommende Semester. Das Verschriftlichen dieses Brainstormings war bereits die erste Herausforderung, denn eine Notiz am eigenen Rechner zu erstellen und dann allen anderen Teammitgliedern beispielsweise per E-Mail zuzuschicken, ist inkonsistent, erzeugt Redundanzen und sorgt von Beginn an für einen unnötig hohen Verwaltungsaufwand und schwindende Motivation. Der erste Schritt war es daher, ein Tool zu finden, über das die angelegten Mitschriften und Notizen in Echtzeit ausgetauscht werden können. Der erste Gedanke war ein im GoogleDrive liegendes, geteiltes Word-Dokument, auf das jeder Zugriff hat. Wenn wir diese Variante gewählt hätten und uns zu sechst ein unschönes, unformatiertes Word-Dokument hätten teilen müssen, müssten wir uns als Media Engineering Studenten jedoch schämen und die Designer in uns bekämen Gänsehaut. Wir entschieden uns daher stattdessen vorausschauend für **Obsidian** (Kapitel 3.1.1).

3.1.1 Obsidian als Notizverwaltungstool

Obsidian ist eine plattformübergreifende Notiz-App, die das von Niklas Luhmann perfektionierte Zettelkastenprinzip in digitaler Form anbietet [1]. In erster Linie ist Obsidian eine Sammlung an Dateien (Notizen), die lokal am Rechner als Markdown Dateien in einem Ordner gespeichert werden und dadurch programmunabhängig und offline verfügbar sind. Durch Obsidian ist es dann möglich, diese Dateien über Hyperlinks, bestehend aus eckigen Klammern, miteinander zu verknüpfen und sich durch die Graphenansicht die bestehenden Verknüpfungen anzeigen zu lassen. Die unten stehende Abbildung zeigt im linken Sechstel die bereits erstellten Notizen, was für einen Überblick sorgt, wer im Team an welchem Problem gearbeitet hat und gegebenenfalls bereits eine Lösung gefunden hat. Die obere linke Hälfte veranschaulicht die Graphenansicht und die Verbindung zwischen den erstellten Notizen. Im unteren linken Bereich wie auch im rechten Bereich sind einzelne Notizen gezeigt, die wir erstellt haben. Beispielhaft wird hier die Anleitung (ReadMe) gezeigt, wie wir im Team mit Obsidian allgemein umgehen und welche Möglichkeiten es bietet. Darüber hinaus ist die Notiz „Github“ im unteren linken Bereich erstellt,

um jedem im Team zu ermöglichen, den geschriebenen Code auszuprobieren oder auch selbst weiterzuentwickeln. Im ganz rechten Sechstel wird einer der größten Vorteile von Obsidian deutlich: die „Unlinked Mentions“, also die unverlinkten Erwähnungen. Wird eine neue Notiz erstellt, ist hier sofort sichtbar, in welchen bereits vorhandenen Notizen der Titel bereits verwendet wird. So kann man sehr einfach vorhandenes Wissen neu verknüpfen und spart sich potenziell doppelte Arbeit. Auch fällt auf, falls eine Notiz zu einem Thema bereits vorhanden ist. Ein weiterer Vorteil von Obsidian ist die Möglichkeit, Plugins zu installieren. Auf diese Weise kann sich jeder aus dem Team seine Arbeitsoberfläche durch das Installieren von Designs selbst gestalten und jederzeit bei höchst möglichem Komfort arbeiten, ohne zwischenzeitlich das Softwaretool wechseln zu müssen. Am Ende sind dann alle projektbezogenen Notizen an einem Ort auffindbar und von allen jederzeit abrufbar.

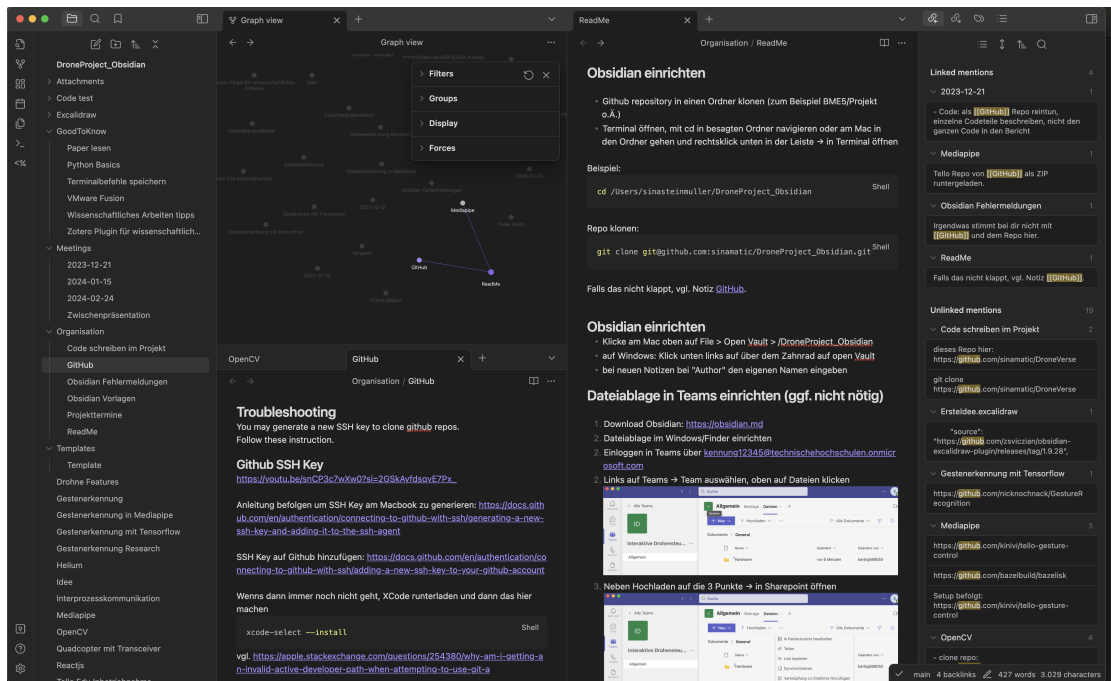


Abbildung 3.1: Screenshot der Obsidian Dokumentation

Die Hauptverwendung von Obsidian ist es, den Fortschritt zu dokumentieren, Besprochenes und Meetings mitzuschreiben und für andere Teammitglieder Anleitungen zu erstellen, die für Reproduzierbarkeit und tiefer gehendes Verständnis sorgen. Damit dies mit größtmöglicher Effizienz funktioniert, werden die Plugins „Templater“, „Github“ und „Excalidraw“ verwendet. **Templater** sorgt dafür, dass jede neu erstellte Notiz ein Metadatenfeld „Author“ anzeigt, welches ausgefüllt werden muss. So werden besitzerlose Notizen vermieden und jeder weiß, an wen er

sich bezüglich weiterer Fragen wenden muss, ohne einen Broadcast in den Chat machen zu müssen. Das Plugin **Github** sorgt dafür, dass unsere Dokumentation automatisch alle 2 Minuten auf **Github** gepusht und von dort gepullt wird, ohne sich um die Versionsverwaltung aktiv kümmern zu müssen. Dies sorgt dafür, dass man jederzeit von jedem Rechner aus am Projekt weiterarbeiten kann. Das Plugin **Excalidraw** ermöglicht es, grafische Skizzen zu erstellen und in die Notizen einzufügen. Dies ermöglicht es Teammitgliedern, die lieber mit einem Stift auf dem Tablet arbeiten wollen, Obsidian ebenfalls zu nutzen. **Hier** kann man sich unserer Obsidian Repository anschauen.

3.1.2 Dokumentenablage in Microsoft Teams

Weil jedoch die Corporate Identity (Kapitel 2) Teil des Projekts ist und hierbei mit großen Bild- und Dokumentdateien gearbeitet werden muss, wurde zusätzlich zur Dokumentation in Obsidian eine Dateiablage auf dem hochschulinternen Microsoft Teams Konto angelegt, auf welchem für Github zu große Projektdateien zentral für alle gespeichert werden. Bei dieser Dokumentablage handelt es sich um ein 1 TB großen Speicherplatz, der sowohl lokal auf dem Rechner als auch online verfügbar ist. Weiterhin verfügt Teams über einen Chat, durch den es uns möglich ist, wichtige Themen zu besprechen und im Nachhinein abrufbar zu machen.

3.1.3 Terminierte Aufgabenverteilung in Trello

Für das eigentliche Projektmanagement eignen sich jedoch weder Microsoft Teams noch Obsidian. Zu diesem Zweck haben wir uns zusätzlich zu diesen beiden Tools für **Trello** entschieden, welches eine Projektmanagement-Anwendung mit vielen Funktionen und installierbaren Plugins ist. In dieser Anwendung lassen sich sehr schnell Aufgaben erstellen, Personen zuweisen und auch Abgabefristen angeben. Neben dem Überblick über Fristen nutzen wir die praktische, unterwegs am Handy verfügbare App auch, um künftige Ideen im Backlog festzuhalten, bei denen wir noch nicht wissen, ob wir diese Funktionen umsetzen werden oder nicht.

Die weiteren Kategorien in Trello haben wir analog unserer Teameinteilung vorgenommen. Neben dem Backlog, welches lediglich eine Ideensammlung enthält, gibt es noch die Kategorien Software, Hardware, Marketing und Organisation. Software beinhaltet alles rund um die Gestenerkennung (Kapitel 3.2) wie auch

die Drohnensteuerung (Kapitel 3.4), Hardware beinhaltet hauptsächlich unsere Quadcopter und Aufgaben an diesen (Kapitel 3.8), Marketing und Organisation sind selbsterklärend.

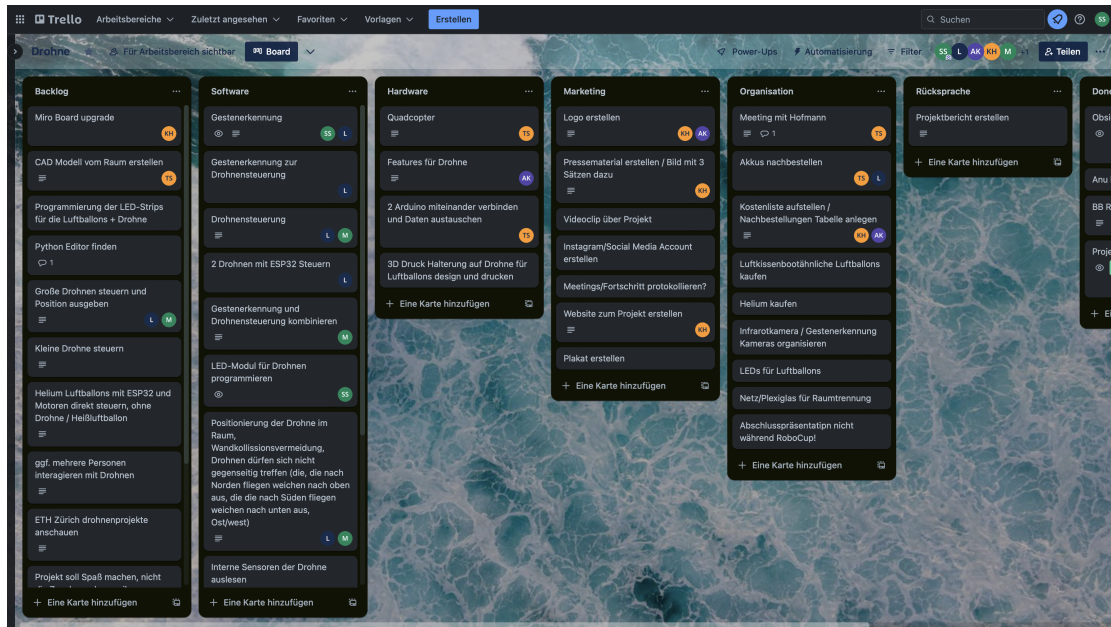


Abbildung 3.2: Screenshot der Trello Aufgabenliste

3.1.4 Visual Studio Code als kostenfreie Entwicklungsumgebung

Um nun mit dem inhaltlichen Arbeiten und Programmieren zu beginnen, wurde sich für Visual Studio Code (VSC) als Entwicklungsumgebung entschieden. Grund hierfür ist es einerseits, dass einige aus dem Team bereits Erfahrung mit VSC haben, es kostenlos ist und bleibt und es viele verschiedene Plugins erlaubt, wodurch man jede Programmiersprache verwenden kann. Da am Anfang noch nicht bekannt war, welche Sprachen benötigt werden, wie die Gestenerkennung bestenfalls umgesetzt werden kann und ob die Drohnen möglicherweise auch anders als über Python angesprochen werden können, sollten so viele Optionen wie möglich offen gehalten werden.

3.1.5 Versionsverwaltung und Codesharing mit Github

Eine passende Entwicklungsumgebung zum Programmieren alleine reicht jedoch noch nicht, um effizient Code zu schreiben. Man benötigt außerdem eine Versionsverwaltungssoftware wie beispielsweise das von uns genutzte Github. Hierfür wurde ein gemeinsames Repository erstellt mit dem Ziel, dass von Beginn an gemeinsam Code geschrieben wird. Dieses „Gemeinsam“ gab es jedoch erst während der geplanten Interprozesskommunikation (Kapitel 3.2.4.1), vorher hat sich jeder selbst um seinen Code und seine Versionsverwaltung gekümmert.

Als Neuling in der Softwareentwicklung Github anzuwenden, brachte schnell einige Herausforderungen mit sich. Angefangen damit, wann und wie oft man einen Branch erstellt und wie genau das Zusammenfügen dieses Branches mit dem Hauptcode dann abläuft. Die Speicherung der oben erwähnten größeren Bilddateien für die Corporate Identity (Kapitel 2) war noch ein verkräftbares Übel. Die versehentlich mehrfach hochgeladenen neuronalen Netze für die Gestenerkennung (Kapitel 3.2) sorgten jedoch leider für eine unwiderrufliche Zerstörung des Repository, weswegen dieses nach vielen misslungenen Rettungsversuchen letztendlich neu erstellt werden musste und die Erfahrung einmal gemacht werden musste.

3.1.6 Overleaf für die gemeinsame Erstellung des Projektberichts

Zu guter Letzt wird **Overleaf** für das gemeinsame Erstellen unseres Projektberichts mittels der Auszeichnungssprache LaTeX genutzt. Dadurch ist es in dem sechsköpfigen Team möglich, in Echtzeit ein einheitliches Layout beizubehalten und den Bericht aufeinander abzustimmen, ohne dass einer aus dem Team am Ende der Arbeit das Design des Dokuments vereinheitlichen muss. Weiterhin ist die Erstellung eines Figurenverzeichnisses, einer Literatursammlung sowie eines Abbildungsverzeichnisses schnell und einfach gemacht, wenn dies gewünscht und benötigt wird.

3.2 Interaktive Drohnensteuerung durch Gestenerkennung

(Sina Steinmüller)

Nachdem das anfängliche Projektmanagement nun eingerichtet war, begann die Arbeit am Projekt. Das folgende Kapitel beschreibt die Planung und Entwicklung der interaktiven Drohnensteuerung. Es gibt viele Möglichkeiten, den Interaktionsaspekt umzusetzen, wie z. B. Sprachsteuerung [16], Gesichtserkennung [43], Körpersteuerung [38] oder die von uns gewählte Gestenerkennung. Die Gestenerkennung ist die intuitivste Art der Drohnensteuerung, [51] bei der man als Anwender am wenigsten überlegen und lernen muss, um diese zu beherrschen. Außerdem ist die Gestenerkennung erweiter- und modifizierbar und wohl auch für alle Besucher am unterhaltsamsten. Nach einem ausführlichen Brainstorming über mögliche zu verwendende Gesten mit dem Ergebnis, dass jeder im Team andere Gesten für besser hält, wurde klar, dass dies ein iterativer Prozess werden würde. Bei diesem wird zu Beginn erst einmal die Technologie erforscht werden, welche Gesten einfach, welche schwer und welche gar nicht umgesetzt werden können.

Grundsätzlich liegt der Fokus darauf, eine rein kamerabasierte Gestenerkennung zu implementieren, die keine zusätzliche Hardwareausstattung wie Datenhandschuhe [49] oder Controller [23] erfordert, weil diese Art der Gestenerkennung nicht nur günstiger und einfacher umzusetzen ist, sondern auch stabiler und verlässlicher ist [51].

Wie bei den meisten Projekten auch war die größte Herausforderung, den Beginn zu finden. Da wir im Team nur 2 Tellodrohnen hatten und eine für das Testen der Steuerung (Kapitel 3.4) und die andere für die Einbindung ins eigene WLAN (Kapitel 3.5) verwendet wurden, erfolgte die Erforschung der Gestenerkennung vorerst drohnenunabhängig. Im Nachhinein betrachtet war dies der größte Fehler des gesamten Projekts, der sowohl den Beginn als unnötig schwierig gestaltete, als auch die schwierige Interprozesskommunikation (Kapitel 3.2.4.1) erforderlich machte. Wir gingen zu diesem Zeitpunkt davon aus, dass es wohl nicht so schwierig sein kann, die von meinem Programm erkannte Geste als Basis für die Steuerung der Drohne zu verwenden. Das war sehr realitätsfern, wie im letzten Abschnitt dieses Kapitels deutlich werden wird.

Durch die große Freiheit bei der Wahl der Programmiersprache und des vielversprechendsten Ansatzes entstand im Laufe der letzten Monate eine große Sammlung an Möglichkeiten zur Implementierung einer Gestenerkennung, die alle verschiedene Vor- und Nachteile haben und unterschiedlich schwierig umzusetzen und weiterzuentwickeln sind. Das Ergebnis dessen ist, abgesehen von einem großen Lerneffekt, eine gute Ausgangsbasis für die Weiterentwicklung einer Gestenerkennung in den nächsten Semestern sowie ein umfangreicher Überblick über verschiedene verwendbare Entwicklungsansätze, die im folgenden beschrieben werden.

3.2.1 Technische Ausstattung

Die folgende Tabelle gibt einen Überblick über die allgemein verwendete technische Ausstattung für die Gestenerkennung, da diese eine große Rolle spielt, wie schnell und flüssig eine Geste erkannt wird und dementsprechend wie schnell die Drohne reagieren wird.

macOS	Sonoma 14.2.1
Prozessor	2,3 GHz 8-Code Intel Core i9
Grafik	Intel UHD-Graphics 630
Speicher	16 GB 2667 MHz DDR4
VMware Fusion	Ubuntu 20.04.5, 64-Bit
IDE	Visual Studio Code (VSC)
VSC Extensions	CMake Tools CMake JavaScript Debugger (Nightly) Pylance Python Python Debugger PlatformIO

Tabelle 3.1: Technische Ausstattung für die Gestenerkennung

3.2.2 Gestenerkennungsbasierte Drohnensteuerung in MediaPipe

Im Projekt von Nikita Kiselov [26] wird Mediapipe verwendet, um eine Gestensteuerung für die Tellodrohne zu realisieren. Dieses Projekt entspricht abgesehen von den konkret verwendeten Gesten genau dem, was unser Projekt werden sollte. Geplant war, das Projekt als Ausgangsbasis zu verwenden und unseren Wünschen entsprechend abzuwandeln oder zu erweitern, was jedoch scheiterte. Nikita selbst hat das Projekt sehr gut dokumentiert und eine eigentlich sehr ausführliche Anleitung geschrieben, wie man es lauffähig machen und erweitern kann. Bedauerlicherweise jedoch verwendet er **Python 3.7**, welches das Lebensende erreicht hat und nicht mehr unterstützt wird. Dennoch war der Wille da, das vorhandene Projekt lauffähig zu machen, also galt es, einige Versuche zu unternehmen, die veraltete Pythonversion zu ersetzen. Neben dem Ansatz, Python 3.7 auf anderen Betriebssystemen zu installieren, wurde außerdem versucht, die Software auf verschiedene Weisen vom Sourcecode aus zu installieren. All diese Ansätze führten zu demselben Ergebnis: Das Programm lässt sich nicht mehr ausführen. Auf Ratschlag von Professor Lano hin, dass das Programm, wenn es so schwierig startet, wohl noch mehr Probleme mit sich bringen würden, selbst wenn wir diese Hürde meistern, ließen wir von dem Versuch ab und beschäftigten uns mit alternativen Entwicklungsansätzen.

3.2.3 OpenCV und C++-Implementierung in VMware Fusion auf Ubuntu 20.04.5

Nach der von Python 3.7 verursachten Frustration und der Angst mit unserem Projekt könnte dasselbe passieren wie mit dem Programm von Nikita, war das Ziel, die Umsetzung der Gestenerkennung so bibliotheksunabhängig wie möglich zu machen. Das nächste Mittel der Wahl war OpenCV, der Open Computer Vision Library, die Algorithmen zur Bildverarbeitung, Objekterkennung, -verfolgung und -analyse bereitstellt. OpenCV ist zwar in C++ geschrieben, bietet aber die Möglichkeit, auch in anderen Programmiersprachen zu schreiben. Da C++ jedoch eine sehr hardwarenahe Programmiersprache ist und deshalb sehr schnell ist, wurde dieser Ansatz gewählt. Gerade zu Beginn war es mir noch nicht klar, dass es schon unheimlich viele verwendbare Lösungen gibt, die ich nutzen kann, und wollte so viel wie möglich selbst machen, um unsere Gestenerkennung vollumfänglich zu

verstehen und erweitern zu können. Da aus den vorherigen Semestern schon eine lauffähige virtuelle Maschine mit Ubuntu eingerichtet war, in der C und C++ flüssig läuft, wurde dort **OpenCV** installiert und die ersten Bildverarbeitungsoperationen für die Gestenerkennung programmiert. Nach dem Kompilieren wurde sofort klar, dass sich die virtuelle Maschine nicht eignet, weil das angezeigte Kamerabild sehr ruckelte und Latenzen aufwies, weswegen eine Echtzeiterkennung von vorn herein schon ausgeschlossen war.

Aus dem Fehler lernend wurde von nun an Mac OS ohne virtuelle Maschine verwendet und versucht, der C++-Code auf dem Mac ausführbar zu machen. Hierbei gab es einige Anlaufschwierigkeiten, weil die OpenCV-Bibliotheken nicht gefunden wurden, CMake bereits in einer anderen Version vorhanden war und nicht kompatibel mit OpenCV war und es mir nur möglich war, den Code lauffähig zu machen, indem das Projekt innerhalb des heruntergeladenen OpenCV-Library-Ordners verschoben wurde. Nach einiger Zeit der Frustration schon zu Beginn des Projektes und dem Scheitern, Beispielprogramme lauffähig zu bekommen, wurde klar, dass eine neue Herangehensweise erforderlich war.

3.2.4 Echtzeit Gestenerkennung mit Tensorflow.js, React.js, Fingerpose und Handpose

Dank der Arbeit zur Echtzeit-Gestenerkennung von Nicholas Renotte [37] stieß ich auf React.js in Kombination mit Tensorflow.js, Fingerpose und Handpose.

React.js ist eine JavaScript Bibliothek, die der Entwicklung von im Browser angezeigten Benutzeroberflächen dient. In der im folgenden beschriebenen Anwendung dient React.js dazu, das Kamerabild in einem Browserfenster anzuzeigen. **Tensorflow.js** dagegen ist eine JavaScript Bibliothek, die es ermöglicht, eigene Maschine-Learning Modelle zu erstellen und zu trainieren. MediaPipe **Handpose** ist eine Bibliothek von Google, die auf maschinellem Lernen basiert und entwickelt wurde, um die Positionen und Orientierungen der Hände in Echtzeit in Bildern oder Videostreams zu erkennen. Außerdem ermöglicht es die präzise Lokalisierung von Gelenkpunkten (Handlandmarks) wie Fingerkuppen, Handgelenken und Zwischengelenken, die dann von **Fingerpose** interpretiert werden, um verschiedene Handgesten zu identifizieren und zu kategorisieren [2]. Genauer gesagt werden durch Handpose 21 Gelenkpunkte identifiziert, von denen jeder Finger von der Fingerspitze an bis zur Hand 4 Gelenkpunkte hat und der Handballen ein eigener Gelenkpunkt ist.



Abbildung 3.3: Handpose Gelenkpunkte

Der folgende Codeausschnitt zeigt Teile des von mir erstellten Hand-Meshs, welches die eben erwähnten Gelenkpunkte in verschiedenen Größen und Farben dargestellt, um die Genauigkeit der Handpose-Gelenkpunkte jederzeit visuell überprüfen zu können. Anders als bei vorgefertigten Meshes, die für jeden Gelenkpunkt die gleiche Größe und Farbe haben, lässt sich hier gut erkennen, welcher Finger von Handpose wie gut erkannt wird oder ob Finger beispielsweise verwechselt wurden.

Listing 3.1: Individueller Fingergelenkstyle der Handposegelenke

```
const fingerjoint_style = {
  [capitate]: { color: "black", size: 15 },

  // Blue Thumb
  1: { color: "#001f3f", size: 10 },
  2: { color: "#003366", size: 8 },
  3: { color: "#004080", size: 6 },
  4: { color: "#66a3ff", size: 4 },
```

```

// Green IndexFinger
5: { color: "#006400", size: 10 },
6: { color: "#008000", size: 8 },
7: { color: "#009900", size: 6 },
8: { color: "#66ff66", size: 4 },

// Purple MiddleFinger
9: { color: "#4b0082", size: 10 },
10: { color: "#6a0dad", size: 8 },
11: { color: "#8a2be2", size: 6 },
12: { color: "#d8bfd8", size: 4 },

// Red RingFinger
13: { color: "#8b0000", size: 10 },
14: { color: "#b22222", size: 8 },
15: { color: "#ff0000", size: 6 },
16: { color: "#ff6666", size: 4 },

// Orange Pinky
17: { color: "#ff4500", size: 10 },
18: { color: "#ff6600", size: 8 },
19: { color: "#ff9933", size: 6 },
20: { color: "#ffd699", size: 4 },
};

```

Abgesehen von dem einfachen Zugriff auf die Gelenkpunkte sorgt Fingerpose dafür, dass man eigene Gesten auch sehr einfach selbst erstellen kann. Der unten gezeigte Code demonstriert dies beispielhaft mit der Geste "Rock", wie sie in dem Spiel "Schere, Stein, Papier" verwendet wird. Der Daumen wird halb gekrümmt definiert mit einer hohen Vertrauenswahrscheinlichkeit von 1.0. Darüber hinaus akzeptiert Fingerpose eine geringere Krümmung des Daumens mit einer Vertrauenswahrscheinlichkeit von 0.5 für den Fall, dass der Daumen nicht in den gekrümmten Fingern steckt, sondern oben drauf liegt. Alle anderen Finger werden als vollständig gekrümmt definiert, wobei sowohl eine volle Krümmung als auch eine halbe Krümmung mit hoher Vertrauenswahrscheinlichkeit vorliegt, jedoch mit etwas geringerer Wahrscheinlichkeit als beim Daumen.

Listing 3.2: Rock Geste mit Fingerpose

```

// thumb: half curled
// accept no curl with a bit lower confidence
rockGesture.addCurl(Finger.Thumb, FingerCurl.HalfCurl, 1.0);
rockGesture.addCurl(Finger.Thumb, FingerCurl.NoCurl, 0.5);

// all other fingers: curled
for (let finger of [Finger.Index,
                    Finger.Middle,
                    Finger.Ring,
                    Finger.Pinky]) {
  rockGesture.addCurl(finger, FingerCurl.FullCurl, 1.0);
  rockGesture.addCurl(finger, FingerCurl.HalfCurl, 0.9);
}

```

Insgesamt ist die React.js Variante sehr beeindruckend und anders als die vorherigen Ansätze vielversprechend und spaßig. Die Bibliothek bietet eine einfache Handhabung und ermöglicht die direkte Darstellung im Browser, was viele Möglichkeiten eröffnet, diese Informationen ansprechend zu gestalten. Besonders beeindruckend sind die Geschwindigkeit und die Genauigkeit der Gestenerkennung. Zufrieden mit den ersten zuverlässig erkannten Gesten ging es als nächsten Schritt darum, mit den vorhandenen Gesten die Drohne zu steuern. Das Repository zu diesem Abschnitt findet sich [hier](#).

3.2.4.1 Interprozesskommunikation

Die Interprozesskommunikation bezeichnet den Versuch, zwischen der Gestenerkennung in React.js mittels JavaScript die Drohnensteuerung (Kapitel 3.4) entsprechend auszuführen. Hierfür sollte die von React.js erkannte Geste als Output dem Pythoncode der Drohnensteuerung als Input dienen, damit für die Zwischenpräsentation eine Gestensteuerung gezeigt werden kann. Dieser Versuch verursachte gleich zwei Probleme: Einerseits ließ sich die Drohne nach aktuellem Stand nur durch das drohneneigene WLAN ansprechen, welches keine Internetverbindung hat. Dem zur Folge musste die React.js-Anwendung offline funktionieren. Grundsätzlich wurde diese Funktion vor Implementierung der Gestenerkennung zufriedenstellend getestet, jedoch nicht mehr nach der Verwendung des neuronalen Netzes Handpose. Es wurde davon ausgegangen, dass die Installation weiterer Module deren vollständigen Download bewirkt, was leider eine Fehleinschätzung war. Die Handpose-Bibliothek

lädt beim Aufruf der Funktion `const net = await handpose.load();` die zugehörigen neuronalen Netze aus dem Internet herunter. Das eigenständige Herunterladen der neuronalen Netze und das Austauschen der Quelle in der Bibliothek schlug leider fehl, sodass sich die App nicht mehr starten ließ. Nach Rücksprache mit Herrn Lano wurde versucht, statt der lokalen Bereitstellung des neuronalen Netzes einen Server zu verwenden, auf dem die Daten liegen. Auch dieser Ansatz blieb leider erfolglos und aufgrund der Kürze der Zeit bis zur Präsentation wurde beschlossen, keine weitere Zeit hierfür zu verschwenden, weil die Drohne ohnehin bald ins eigene WLAN eingebunden werden wird und die Offlinefunktion dann nicht mehr nötig ist.

Der nächste Schritt war es dann, die erkannte Geste online an ein Pythonskript zu schicken. Der erste Versuch war es, diese Datenübergabe durch die Simulation eines Tastendruckes zu simulieren. Die React-Anwendung sollte bei einer erkannten Geste eine Taste drücken, welche dann vom Pythonskript als Flugbefehl interpretiert werden sollte. Dies klappte jedoch leider nicht. React gab zwar aus, es würde eine Taste gedrückt werden, jedoch wurde dieser Tastendruck im restlichen Betriebssystem nicht erkannt. Man konnte also weder ein Textdokument beschreiben, noch den Input für Flugbefehle verwenden. Weitergehende Recherche machte deutlich, dass dies aufgrund der Browsersicherheit gewünscht ist. Anschließend wurde parallel an zwei neuen Ansätzen gearbeitet: einerseits die standardmäßige Verwendung eines Servers, über den der Datenaustausch stattfindet und andererseits der direkte Aufruf eines Pythonskripts innerhalb der React-Anwendung mittels Pyodide.

Die Einrichtung eines Servers, an den Daten von React.js gesendet und von dem Daten für die Drohnensteuerung empfangen werden können, gestaltete sich in der Kürze der Zeit als zu schwierig. Wir testeten Ansätze mit Java und Python, benötigen aber noch einige Recherche, wie genau man Daten von und an einen Server senden kann.

Der Aufruf eines Python Skripts innerhalb einer React.js Anwendung dagegen gelang nach einiger Recherche unerwartet einfach. Das Projekt **Pyodide** ermöglicht es, Python-Code in Webanwendungen auszuführen, indem es eine vollständige Python-Laufzeitumgebung in WebAssembly kompiliert. So können Entwickler Python-Code direkt im Browser auszuführen, ohne dass eine separate Python-Installation auf dem Server oder Client erforderlich ist.

Die Implementierung des Pythonskripts gelang mithilfe des Youtube Tutorials von [AkashBhat](#). Der folgende Codeausschnitt zeigt die erfolgreiche Ausführung des Pythonskripts main.py in der React.js Anwendung. Testweise erfolgt hier nur als Ausgabe nur ein Python Printbefehl, welcher jedoch künftig durch die konkrete Steuerung der Drohne ersetzt werden kann.

Listing 3.3: Pythonintegration in React.js

```
import script from "./python/main.py";

( Teile fehlen )

const runScript = async (code) => {
  if (!pyodideInstance) {
    pyodideInstance = await window.loadPyodide({
      indexURL:
        "https://cdn.jsdelivr.net/pyodide/v0.18.1/full/",
    });
  }

  return await pyodideInstance.runPythonAsync(code);
};

( Teile fehlen )

async function handleDetectedGesture(hand) {
  const estimatedGesture = await estimateGesture(hand);
  handleEstimatedGesture(estimatedGesture);

  if (estimatedGesture.gestures.length > 0) {
    const { name } = estimatedGesture.gestures[0];
    if (name === "thumbs_up") {
      const pythonCode = `
        result = thumbs_up()
        result
      `;
      const result = await runScript(pythonCode);
      console.log('Python result: ${result}');
    }
  }
}
```


}

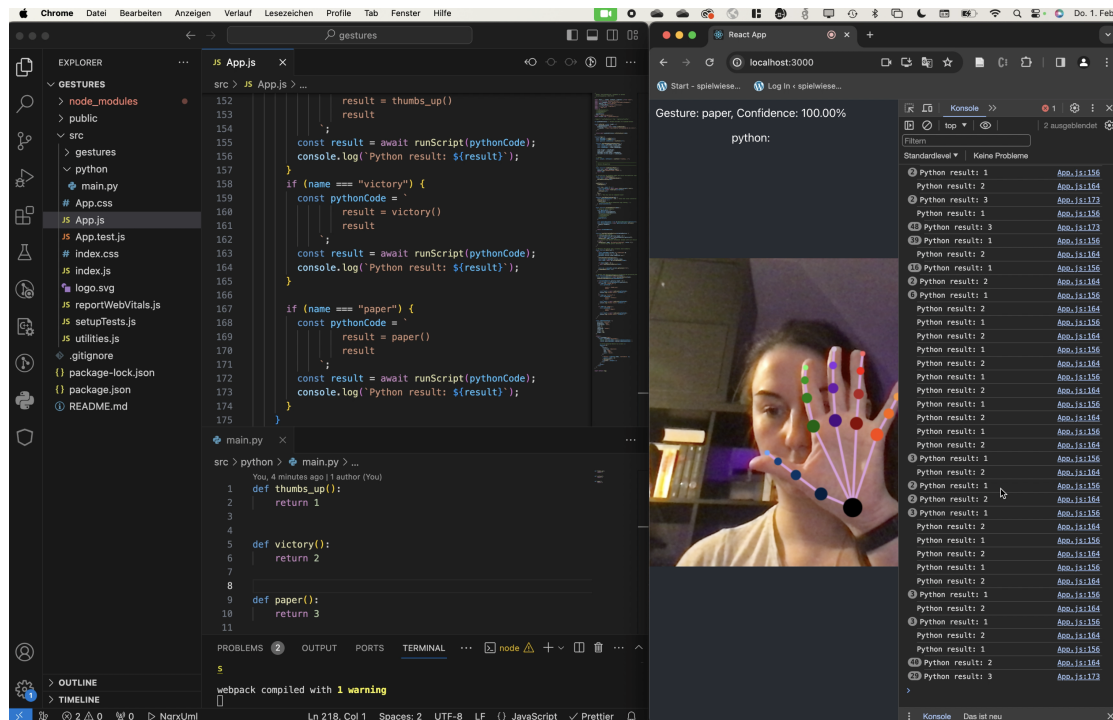


Abbildung 3.4: Ausgabe eines Pythonskripts in einer React.js Anwendung

Sowohl der Ansatz des Servers als auch die Ausführung der Pythonfunktionen in React.js sind vielversprechend und werden im folgenden Projektsemester weiterentwickelt. Aufgrund unseres überarbeitungsbedürftigen Zeitmanagements jedoch musste für die Zwischenpräsentation ein neuer Ansatz umgesetzt werden, der keine Interprozesskommunikation erfordert.

3.2.5 Python mit TensorFlow

Die letzte verfolgte Strategie ist die Implementierung einer Gestenerkennung direkt in Python ohne Interprozesskommunikation (Kapitel 3.2.4.1). Aufgrund der gesammelten Informationen über die Zuverlässigkeit von Tensorflow in React.js wurde eine vorhandene Gestenerkennung gesucht, die bereits ein fertiges neuronales Netz verwendet. Das Tutorial von TechVidvan [3] wurde als Ausgangsbasis verwendet und um die bereits vorhandene Drohnensteuerung (Kapitel 3.4) erweitert. Zufriedenstellend ist diese Lösung jedoch nicht, weswegen sie lediglich für die Zwischenpräsentation als Proof-of-Concept diente. Es ist auf diese Weise zwar möglich,

die vorgefertigten 10 Gesten zuverlässig zu erkennen, eigene Gesten zu erstellen ist jedoch nicht möglich. Darüber hinaus soll weiterhin an der Gestenerkennung und der Drohnensteuerung unabhängig voneinander entwickelt werden, weil dies die Aufgabenteilung wie auch die Versionsverwaltung erleichtert. Für die Präsentation konnten wir jedoch eine einfache Anwendung erstellen. Die Gestenerkennung funktionierte recht gut, allerdings stießen wir auf das Problem, dass der Befehl zum Abheben der Drohne zu oft und zu schnell gesendet wurde, was manchmal zu Fehlern führte. Dies konnte jedoch durch den Auto-Start der Drohne beim Programmstart erfolgreich gelöst werden. Das zugehörige Repository ist [hier](#) zu finden.

3.2.6 Resümee und Ausblick

Zusammenfassend gibt es zwei weiterentwicklungswerte Ansätze für die Gestenerkennung. Ungeachtet der persönlichen Abneigung Python gegenüber ließ sich auf diese Weise eine schnelle und präzise Gestenerkennung mit zu Grunde liegendem Tensorflow realisieren. Der Vorteil der nicht mehr nötigen Interprozesskommunikation lässt sich nicht abstreiten. Genauso eignet sich die React.js Gestenerkennung für die Weiterentwicklung, auch wenn dies einmalig die Entwicklung einer Interprozesskommunikation erforderlich macht, die dann jedoch vermutlich für einige zukünftige Projekte verwendet werden kann.

Eine weitere Möglichkeit, die untersucht werden sollte, ist die Verwendung von ROS für die Interprozesskommunikation [52]. **ROS** (Robot Operating System) ist ein Open-Source-Roboter-Entwicklungsframework, das eine Vielzahl von Werkzeugen, Bibliotheken und Konventionen für den Bau von Robotersystemen bereitstellt und verschiedene Programmiersprachen wie C++, Python und mehr unterstützt. ROS bietet außerdem eine umfangreiche Sammlung von Paketen für Funktionen wie Sensorverarbeitung, Navigation und Steuerung und ist der aktuelle Stand der Technik in der Entwicklung von autonomen Roboter- und Drohnenlösungen [36]. Bisher hatte die Recherche zur Verwendung von ROS eine geringe Priorität, hauptsächlich wegen des nötigen Einrichtungsaufwandes. ROS wird nur von Linux Distributionen unterstützt und die Verwendung einer virtuellen Maschine ist wie bereits erwähnt aufgrund der Kameralatenzen keine Option. Darüber hinaus ist die Verwendung von ROS für unsere Projektarbeit nach Aussage von Professor Lano möglicherweise „Overkill“. Zu guter Letzt ist die Definition und Implementierung

von passenden Gesten für die Drohnensteuerung noch auszubauen. Möglicherweise bietet es sich hier an, ein eigenes neuronales Netz zu erstellen und mit Daten zu füttern.

3.3 Technische Komponenten der Tello Drohne

(Maximilian Richter)

Am Anfang des Projekts wurden uns von Professor Lano neben den in Kapitel 3.8 erwähnten Quadcoptern auch zwei Tello-Drohnen vom Hersteller DJI zur Verfügung gestellt. Diese eignen sich aufgrund ihrer weit entwickelten Open Source Bibliotheken, der Robustheit und nicht zuletzt ihres vergleichsweise geringen Preises hervorragend für studentische Projekte [20].

Zunächst galt es aber, die Spezifikationen der Drohne mit den Anforderungen für das Projekt zu vergleichen.

Folgende Komponenten sind in der Tello Drohne verbaut [46]:

- HD-Kamera
- Barometer
- Ultraschallsensor
- Netzwerk-Chip (802.11n Wi-Fi, 2,4 GHz)
- Status LED
- Akku

Im Folgenden wird auf die einzelnen Bauteile in Bezug auf unser Projekt eingegangen.

3.3.1 HD Kamera

Die Kamera der Tello-Drohne nimmt Fotos mit 5 Megapixel auf, hat ein Sichtfeld von $82,6^\circ$ und überträgt Videos in HD 720p [46]. Aufgrund des Ziels, mit der Drohne zu interagieren, ist diese integrierte Kamera für das Projekt unabdingbar. Weil zunächst ein Proof of Concept für die erfolgreiche Gestensteuerung wichtig war, verzichteten wir jedoch vorerst darauf, die Videodaten der Drohne für die Gestenerkennung zu analysieren.

Als vereinfachte Alternative wurde die Webcam des steuernden Laptops verwendet.

3.3.2 Barometer

Das Barometer wird verwendet, um die Höhe der Drohne im Raum festzustellen. Dafür misst der Luftdrucksensor den sich verändernden Luftdruck, stellt einen Bezug zwischen Luftdruck und Höhe her und übermittelt den resultierenden Höhenunterschied an die Drohne.

3.3.3 Ultraschallsensor

Der Ultraschallsensor ist an der Unterseite der Drohne angebracht und findet seine Anwendung in der Abstandsmessung.

Das ist wichtig, um den Abstand der Drohne zu Gegenständen unter ihr feststellen zu können. Dafür sendet der Sensor einen hochfrequenten Schallimpuls aus, welcher von Oberflächen reflektiert wird. Anschließend wird anhand der Zeit, die der Schall benötigt, um wieder zum Sensor zu gelangen, der Abstand berechnet.

3.3.4 Netzwerkchip

Die Verbindung zwischen Tello-Drohne und PC erfolgt per WLAN auf dem 2,4 GHz Frequenzband [46].

Schon früh war klar, dass dieses Detail zu Problemen führen könnte.

Die meisten Geräte (Handys, Laptops usw.) benutzen nämlich eben jenes Frequenzband für ihre WLAN-Verbindung. Wenn nun bei einer Präsentation der Drohne viele Personen mit eingeschaltetem WLAN anwesend wären, könnte dies durch Überlastung des Frequenzbands zu Latenzen bei der Steuerung der Drohne führen. Lösen lässt sich dieses Problem dennoch recht einfach, indem wir Besucher unserer Präsentation bitten, ihre mobilen Geräte in den Flug-Modus bzw. auszuschalten.

3.3.5 Status LED

Frontal an der Drohne ist eine Status LED angebracht.

Obwohl sie für das Projekt nicht zwingend notwendig ist, bietet uns diese LED Möglichkeiten, das Ausführen von Befehlen transparenter zu gestalten.

Es ist geplant, bestimmte Farben spezifischen Steuerungsbefehlen der Drohne zuzuordnen. So bekommen Anwender von der Drohne Echtzeit-Feedback, welche Geste im Moment erkannt wird bzw. welcher Befehl ausgeführt wird.

Eine mögliche intuitive Farbzuzuordnung der Standardkommandos wäre beispielsweise:

- Blau (Himmel) - Drohne fliegt nach oben
- Grün (Boden) - Drohne fliegt nach unten
- Rot (Anfangsbuchstabe R) - Drohne fliegt nach rechts
- Lila (Anfangsbuchstabe L) - Drohne fliegt nach links

3.3.6 Akku

Die Tello-Drohne verfügt über einen Akku mit einer Kapazität von 1,1 Ah bei 3,8V Spannung. Aufgeladen wird dieser über einen Micro-USB Anschluss.

Laut Hersteller reicht die Kapazität für eine durchgehende Flugzeit von 13 Minuten. Glücklicherweise ist der Akku herausnehmbar, weshalb auch längere Präsentationen unseres Projekts mit mehreren Akkus durchführbar sind [\[46\]](#).

3.4 Steuerung der Tello Drohne

Nachdem die technischen Spezifikationen der Drohne geklärt und für gut befunden wurden, konnten nur erste Ansätze zur Steuerung gemacht werden.

Vorerst haben wir die Stabilität des Flugverhaltens über eine für Handys programmierte App zur Steuerung der Drohne getestet. Hierbei gab es keine Argumente, die gegen eine Verwendung des Modells sprachen.

Eine weitere Steuermethode ist die Verwendung der Programmiersprache Scratch. Da diese jedoch für Schüler konzipiert ist und daher über sehr begrenzte Möglichkeiten verfügt, wird für das Projekt zur Steuerung der Drohne die Programmiersprache Python verwendet.

3.4.1 Bibliothek

Wie schon im vorherigen Kapitel erwähnt, gibt es für die Tello-Drohne weitreichende Python Bibliotheken. Verwendet wird für das Projekt die Open-Source Bibliothek „DJITelloPy“ von Damià Fuentes Escoté [8].

Installiert wird diese Bibliothek über den Kommandozeilenbefehl:

- `pip install djitellopy`

Für eine Installation, bei der die Bibliothek nach den eigenen Wünschen anpassbar sein soll, benötigt man die Kommandozeilenbefehle:

- `git clone https://github.com/damiafuentes/DJITelloPy.git`
- `cd DJITelloPy`
- `pip install -e .`

3.4.2 Gängige Methoden

Durch die DJITelloPy Bibliothek werden viele Methoden bereitgestellt.

Die wichtigsten sind im Folgenden zusammen mit ihrer Funktion aufgelistet [9].

- `DroneVerse = tello.Tello()`:
Instanziierung der Tello Klasse mit dem Namen „DroneVerse“.
- `DroneVerse.connect()`:
Wechseln in den SDK Modus - muss vor Steuerbefehlen einmalig aufgerufen werden.
- `DroneVerse.takeoff()`:
Zum Starten der Drohne - Drohne fliegt nach oben und hält die Position.
- `DroneVerse.move_up(x)`: Drohne fliegt um „x“ cm nach oben.
- `DroneVerse.move_down(x)`: Drohne fliegt um „x“ cm nach unten.
- `DroneVerse.move_forward(x)`: Drohne fliegt um „x“ cm nach vorne.
- `DroneVerse.move_back(x)`: Drohne fliegt um „x“ cm nach hinten.
- `DroneVerse.move_right(x)`: Drohne fliegt um „x“ cm nach rechts.
- `DroneVerse.move_left(x)`: Drohne fliegt um „x“ cm nach links.
- `DroneVerse.rotate_clockwise(x)`:
Drohne rotiert „x“ Grad im Uhrzeigersinn.
- `Droneverse.rotate_counter_clockwise(x)`:
Drohne rotiert „x“ Grad gegen den Uhrzeigersinn
- `DroneVerse.land()`:
Drohne landet
- `DroneVerse.get_battery()`:
Gibt den Akkustand der Drohne als Int in % zurück
- `DroneVerse.get_height()`:
Gibt die Höhe als Int in cm zurück

3.4.3 Tastatur-Steuerung

Nachdem eine funktionierende Gestenerkennung programmiert wurde, bei der für erkannte Gesten ein Tastendruck simuliert wird (siehe Kapitel 3.2.4.1), galt es eine Steuerung für die Drohne, basierend auf dem Tastatur-Input, zu realisieren.

Verwendet wurde dafür die Open-Source Bibliothek „pygame“ [13][12].

Benötigt wurden zwei Module:

- TastenErkennung.py
- DrohnenSteuerung.py

Listing 3.4: Modul zur Tastenerkennung - TastenErkennung.py

```
import pygame
#Pygame Initialisierung bzw. Tastenerkennung
def init ():
    pygame.init()
    win = pygame.display.set_mode((400,400))

def getKey(keyName):
    ans = False
    for eve in pygame.event.get(): pass
    keyInput = pygame.key.get_pressed()
    myKey = getattr(pygame, 'K_{}'.format(keyName))
    if keyInput[myKey]:
        ans = True
    pygame.display.update()
    return ans
#Ausgabe zur Ueberpruefung der Funktionalitaet
def main():
    if getKey("w"):
        print("Taste w gedrueckt")
    if getKey("s"):
        print("Taste s gedrueckt")

if __name__ == '__main__':
    init()
    while True:
        main()
```

Listing 3.5: Modul zur Steuerung der Drohne - DrohnenSteuerung.py

```
from djitellopy import tello
from time import sleep
import Tastenerkennung as te

# Initiieren der Tastenerkennung
te.init()
# Instanziierung der Drohne, Verbinden der Drohne und Ausgabe des
  Batteriestands
droneverse = tello.Tello()
droneverse.connect()
print(droneverse.get_battery)

# Tastenzuweisung
def getKeyboardInput():
    lr, fb, ud, yv = 0, 0, 0, 0
    speed = 50

    if te.getKey("w"):
        ud = speed #hoch
    if te.getKey("s"):
        ud = -speed #runter
    #Platz fuer weitere Tastenzuweisungen

    return [lr, fb, ud, yv]

# "Gameloop"
while True:
    vals = getKeyboardInput()
    droneverse.send_rc_control(vals[0], vals[1], vals[2], vals[3])
    sleep(0.05)
```

Damit der repräsentative Code-Ausschnitt übersichtlich bleibt, wurden die Tastenzuweisungen für links und rechts (lr), vorwärts und zurück (fb) und die beiden Rotationsrichtungen (yv) absichtlich weggelassen.

Mithilfe dieser Module lässt sich die Drohne nun über den Tastatur-Input steuern, allerdings galt es noch, den in der Gestenerkennung simulierten Tastendruck an dieses Python Programm zu übergeben. (siehe Kapitel [3.2.4.1](#))

3.4.4 Steuerungs-Ablauf

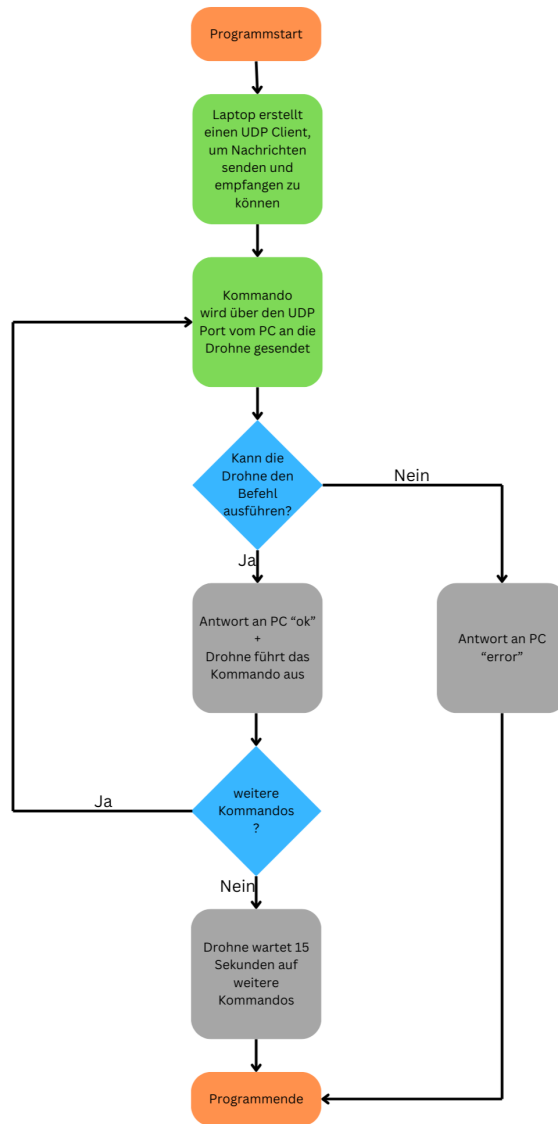


Abbildung 3.5: Flowchart Ablauf Drohnensteuerung

3.5 Ansteuerung mehrerer Tello-Drohnen

(Lisa Berbig)

Das Ziel am Ende dieser Projektarbeit ist es nicht nur eine, sondern mehrere Drohnen gleichzeitig fliegen zu lassen und steuern zu können. Um die Drohnen untereinander zu koordinieren, ist es notwendig, alle gemeinsam über einen Computer zu steuern.

Mehrere gleichzeitig in der Luft fliegende Drohnen werden im Folgenden auch als Drohnenschwarm bezeichnet.

3.5.1 Problembeschreibung

Wie bereits im Kapitel 3.3.4 erläutert, wird die Tello-Drohne über WLAN mit dem Computer, der sie steuert, verbunden. Jede Tello-Drohne baut dafür ihr eigenes WLAN-Netzwerk auf, mit dem man sich dann mit einem WLAN fähigem Gerät verbindet, um sie zu steuern. Da ein Computer sich jedoch normalerweise nur mit einem WLAN-Netzwerk gleichzeitig verbinden kann, können nicht einfach so zwei oder mehrere Drohnen mit einem Computer verbunden werden.

Es musste also eine Lösung gefunden werden, die es erlaubt, mehrere Drohnen über einen Computer zu steuern. Dazu wurden verschiedene Lösungsansätze betrachtet, welche im Folgenden genauer erläutert werden sollen.

3.5.2 Lösungsansatz 1: Zusammenführen der WLAN-Netzwerke

Bereits zu Beginn der Recherche zu diesem Thema war klar, dass es sinnvoll wäre, alle Drohnen in einem gemeinsamen Netzwerk zu haben.

Da die Drohnen jedoch alle ihre eigenen Netzwerke aufbauen, müsste man diese auf irgendeine Weise in einem übergeordneten Netzwerk zusammenfassen. Dies führte zunächst einmal zu der Frage, ob das technisch überhaupt möglich ist. Dass im späteren Verlauf meiner Recherche eine viel einfachere Lösung gefunden werden würde, die es erlaubt, alle Drohnen in einem Netzwerk zu haben, war zu diesem Zeitpunkt noch nicht absehbar.

Zwischen Drohne und Computer muss in beide Richtungen kommuniziert werden können, also genügt es nicht nur die Netzwerke der Drohnen zusammenzufassen, sondern sie müssen auch verwaltet werden, um die Drohnen von diesem zentralen Netzwerk ausgehend auch einzeln ansteuern zu können. Die Recherche nach solch einer Lösung verlief nicht sehr erfolgreich, weshalb schon bald andere Möglichkeiten in Betracht gezogen wurden.

3.5.3 Lösungsansatz 2: Promiscuous-Modus

Von Professor Hopf wurde im späteren Verlauf des Semesters vorgeschlagen, es mit dem Promiscuous-Modus zu versuchen. Der Promiscuous-Modus ermöglicht es, sich mit mehreren WLAN-Netzwerken zeitgleich zu verbinden. So könnte also ein Computer mit den Netzwerken mehrere Drohnen auf einmal verbunden werden. Da sich mit dieser Lösungsidee, wie bereits erwähnt, erst gegen Ende des Semesters beschäftigten und kurz darauf auch eine noch bessere Lösung gefunden wurde, wurde dieser Ansatz nicht lange weiter verfolgt.

3.5.4 Lösungsansatz 3: ESP32

Die weitere Recherche nach einer Möglichkeit, mehrere Tello-Drohnen gleichzeitig zu steuern, führte zu einer Lösung, die die Drohnen mit mehreren ESP32 steuert.

Der ESP32, im Folgenden auch ESP genannt, ist ein Mikrocontroller, der für verschiedenste IoT-Anwendungen genutzt werden kann. Über USB an einen Computer angeschlossen, bietet er verschiedenste Peripherieschnittstellen, wie beispielsweise WLAN, Bluetooth und diverse Analog-digital-Umsetzer. Für unser Projekt stand uns das ESP32 NodeMCU Development Board auf Basis des ESP-WROOM-32 Chipsatzes zur Verfügung. [\[50\]](#) [\[45\]](#)

Um einen Drohnenschwarm zu steuern, benötigt man für jede Drohne einen ESP, welche man dann über USB an steuernden Computer anschließen kann. In dem Python-Programm zur Drohnensteuerung weißt man dann jeder Drohne einen ESP zu, über welchen man diese dann ansteuern kann.

Im Internet konnte ein GIT-Repository gefunden werden, welches das bereits implementierte. Also sollte dieses zunächst einmal ausprobiert werden. [\[21\]](#)

Um dieses Programm mit unserer Drohne und unserem ESP zu verwenden, muss in der ESPTelloCLI.py der serial port Name des ESP32 und die SSID der Tello-Drohne einzutragen werden.

Anschließend sollte es möglich sein, die Drohne über den ESP zu steuern. Bei der Ausführung des Programmes wurde jedoch die Exception „connect failed“ geworfen. Durch Debugging des Codes konnte herausgefunden werden, dass dies in der swarm.py in Zeile 42 passiert, nachdem der connect-Befehl aus der tello.py aufgerufen wurde. Was genau der Grund dafür ist, konnte auch nach längerer Recherche nicht herausgefunden werden. Die Verbindung zwischen Computer und ESP funktioniert problemlos (der ESP blinkt beim Ausführen des Programmes kurz auf), jedoch scheitert es an der Verbindung zwischen dem ESP und der Drohne.

Nachdem zu diesem Zeitpunkt erfolglos einige Zeit in die Steuerung der Drohne mit dem ESP investiert wurde, wurde nach einem anderen Lösungsansatz gesucht.

3.5.5 Lösungsansatz 4: Drohnen in ein externes Netzwerk aufnehmen

Nachdem sich der ESP als kein optimaler Ansatz herausstellte, konnte nach weiterer Recherche noch eine weitere Möglichkeit zur Lösung des Problems gefunden werden.

Die EDU-Version der Tello-Drohne, welche uns für unsere Projektarbeit zur Verfügung gestellt wurde, besitzt die Funktion, sich mit einem bereits vorhandenen Netzwerk zu verbinden, anstatt ihr eigenes Netzwerk aufzubauen.[9] Da dies ausschließlich bei der Tello EDU möglich ist, hatte ich diese Funktion zu Beginn der Recherche nicht gefunden.

Diese Funktion der Drohne ermöglicht es also, sowohl mehr Drohnen, also auch den Computer, der die Drohnen steuert, in unser eigenes Netzwerk aufzunehmen und somit alle Komponenten der Drohnensteuerung in einem gemeinsamen Netzwerk zu haben.

Das bietet nicht nur den Vorteil, dass nicht mehr mehrere verschiedene Netzwerke koordiniert werden müssen, sondern auch, dass die Drohnen selbst kein Netzwerk mehr aufbauen müssen. Denn dadurch, dass die Drohnen jetzt nur noch ein Client

in einem anderen Netzwerk sind, anstatt selbst ein Netzwerk aufzubauen, sollte sich auch ihre Akkulaufzeit etwas verbessern.

Dieses Semester gab es außerdem das Problem, dass bei zu vielen WLAN-fähigen Geräten im Raum die 2,4 GHz Frequenz, auf der die Drohne kommuniziert, zu sehr ausgelastet ist und das Signal der Drohne dadurch zu schwach war (vgl. 3.3.4). Wenn zukünftig aber das Netzwerk nicht mehr über die Drohne, sondern über einen Router aufgebaut wird, kann durch die Wahl eines stärkeren Routers unser Netzwerk besser gegen andere Endgeräte im Raum ankommen.

3.5.6 Fazit zur Steuerung mehrere Drohnen

Schließlich hat sich der 4. Lösungsansatz als der für dieses Projekt sinnvollste und am besten umsetzbare herausgestellt. Nicht nur ist dies die übersichtlichste und aufwandsärmste Lösung, sie bietet außerdem den Vorteil, dass eine leichte Verbesserung der Akkulaufzeit zu erwarten ist und das Problem der instabilen Verbindung zwischen Drohen und Computer durch einen stärkeren Router verhindert werden kann.

Für das nächste Semester ist also geplant, an dieser Lösungsidee weiterzuarbeiten, diese praktisch zu testen und zu optimieren.

3.6 Kollisionsverhinderung

(Lisa Berbig)

Drohnen sind Objekte, welche sich eigenständig in der Luft bewegen und stellen dadurch prinzipiell erst einmal eine Gefahr dar. Um diese möglichst gering zu halten, ist es wichtig, mögliche Kollisionen einer Drohne mit ihrer Umgebung oder einer anderen Drohne frühzeitig zu erkennen und somit verhindern zu können.

3.6.1 Kollisionsverhinderung mit der Umwelt

Um sicherzustellen, dass die Drohnen nicht mit den Besuchern unserer Installation zusammenstoßen, wird der Flugbereich der Drohnen zunächst einmal mit einem Netz abgegrenzt. Dadurch ist dann auch der Flugbereich der Drohnen begrenzt, was auch eine Kollisionsverhinderung mit der restlichen Umwelt erleichtert.

Durch das Festlegen eines festen Startpunktes für jede Drohne kann nachvollzogen werden, wo im Raum sich die Drohne befindet. Mit Hilfe des Protokolls der Gestensteuerung kann man dann berechnen, wo sich die Drohne im Raum befindet.

Außerdem hat die Tello-Drohne einen eingebauten Hözensensor, durch den zusätzlich auch noch die Höhe der Drohnen bekannt ist (vgl. 3.3.2). Bevor die Drohne sich also in eine Richtung bewegt, wird zuvor geprüft, ob sie mit dieser Bewegung über den vorher festgelegten Bereich hinausfliegen würde. Wenn dies der Fall wäre, kann die Bewegung der Drohne verhindert werden.

3.6.2 Kollisionsverhinderung der Drohnen untereinander

Ein Schwarm an Drohnen in der Luft birgt immer auch die Gefahr, dass die Drohnen untereinander zusammenstoßen können, besonders wenn die Drohnen unabhängig voneinander durch Gesten gesteuert werden. Deshalb ist es wichtig zu wissen, wo im Raum sich die einzelnen Drohnen befinden, um erkennen zu können, wenn sich zwei oder mehrere Drohnen auf Kollisionskurs befinden.

Die Kollisionsverhinderung der Drohnen untereinander ist deutlich komplexer als die Kollisionsverhinderung mit der Umwelt, da man es hier mit mehreren sich im dreidimensionalen Raum bewegendem Objekte zu tun hat.

Um dies umzusetzen, kann man den Raum, in dem sich die Drohnen bewegen, in eine vertikale Achse und eine horizontale Ebene aufteilen.

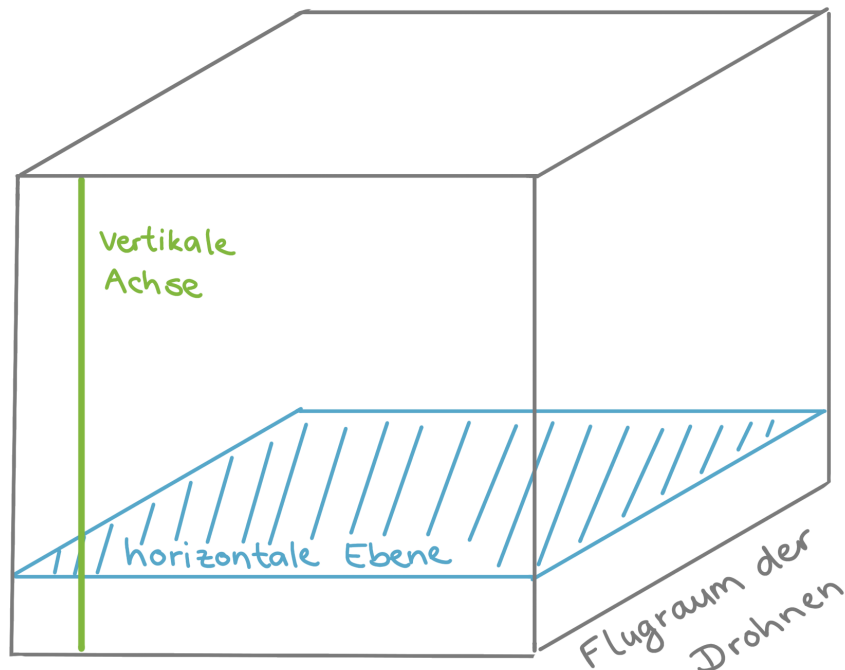


Abbildung 3.6: Skizze zur Raumaufteilung für die Kollisionserkennung

Die vertikale Achse beschreibt, auf welcher Höhe sich die Drohne befindet, während die horizontale Ebene Auskunft über die Position der Drohnen parallel zum Boden gibt.

3.6.2.1 Positionserkennung auf der vertikalen Achse

Bei der Positionserkennung auf der vertikalen Achse wird überprüft, auf welcher Höhe sich die Drohnen befinden. Das ist bei der Tello-Drohne relativ einfach. Dank des eingebauten Hözensensors weiß die Drohne jederzeit, auf welcher Höhe sie sich befindet.

Diese Angabe ist zwar nicht zentimetergenau, doch mit Einberechnung einer Toleranz ausreichend genau. Wie hoch genau diese Toleranz sein muss, werden wir im Laufe des nächsten Semesters testen. Dabei muss auch beachtet werden, ab welcher Distanz sich zwei Drohnen das Flugverhalten der jeweils anderen Drohne durch den Luftstrom ihrer Propeller beeinflussen.

Die Höhendaten der Drohnen können dann zur vertikalen Positionserkennung miteinander verglichen werden. Stellt sich dabei heraus, dass zwei Drohnen sich in etwa auf derselben Höhe befinden, überprüft man deren Position auf der horizontalen Ebene.

3.6.2.2 Positionserkennung auf der horizontalen Ebene

Wenn sich nun also zwei Drohnen in etwa auf der gleichen vertikalen Ebene befinden, muss nun überprüft werden, ob sie sich auch auf der horizontalen Ebene zu nahe kommen. Hierfür kann auf Methoden der Bildverarbeitung zurückgegriffen, welche wir im 3. Semester in Interaktion kennengelernt haben.

Dafür wird mit einer Kamera gearbeitet, die den Bereich, in dem die Drohnen fliegen, von oben filmt.

Um an dieser Stelle noch einmal kurz daran zu erinnern: Das Ganze soll in einem relativ dunklen Raum stattfinden und auf den Drohnen werden sich Luftballons befinden, welche in verschiedenen Farben leuchten.

Die Dunkelheit des Raumes und die farbig leuchtenden Ballons kann man sich nun bei der Positionserkennung in der horizontalen Ebene zu Nutzen machen.

Mit Hilfe von OpenCV kann man Algorithmen der Bildverarbeitung auf die Videoausgabe der Kamera anwenden. Dafür wird der Farbraum zunächst von RGB in den HSV-Farbraum umgewandelt. HSV beschreibt eine Farbe durch ihren Winkel im Farbkreis, ihrer Sättigung und ihrer Helligkeit. Das erleichtert es uns, die Farbwerte festzulegen, die aus dem Bild extrahiert werden sollen. Für die Erkennung eines blauen Ballons würde man sich also beispielsweise die Farbwerte von HSV(100, 50, 50) bis HSV(130, 255, 255) herausnehmen. Diese Werte werden als untere bzw. obere Grenze festgelegt. Mit diesen Informationen kann nun eine Maske erstellt werden, welche nur die Bildpunkte im Bereich der vorher festgelegten Farbwerte durchlässt. Schließlich wird auf das Bild eine Bitweise UND Operation ausgeführt. Dadurch sind nur noch die sich im blauen Farbbereich befindenden Pixel sichtbar.

Da es sich hier um ein Videobild handelt, muss dieser Algorithmus natürlich für jedes Bild wiederholt werden. Hierfür wird eine While-Schleife mit der Bedingung „True“ verwendet. Dabei ist es wichtig, auch eine Abbruchbedingung für die Schleife, zum Beispiel in Form eines Tastendrucks, zu implementieren.

Auf diese Weise kann dann also bestimmt werden, wo sich die Drohnen auf der horizontalen Ebene befinden. Durch die unterschiedlichen Farben der Ballons kann außerdem zugeordnet werden, wo im Bild welche Drohne ist.

Im Folgendem ist beispielhaft zu erkennen, wie das für einen blauen Ballon aussehen würde:

Listing 3.6: Programm zur Erkennung blauer Farbbereiche in einem Video

```
import numpy as np
import cv2

cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    width = int(cap.get(3))
    height = int(cap.get(4))

    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    lower_blue = np.array([100, 50, 50])
    upper_blue = np.array([130, 255, 255])

    mask = cv2.inRange(hsv, lower_blue, upper_blue)

    result = cv2.bitwise_and(frame, frame, mask=mask)

    cv2.imshow('frame', result)

    if cv2.waitKey(1) == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

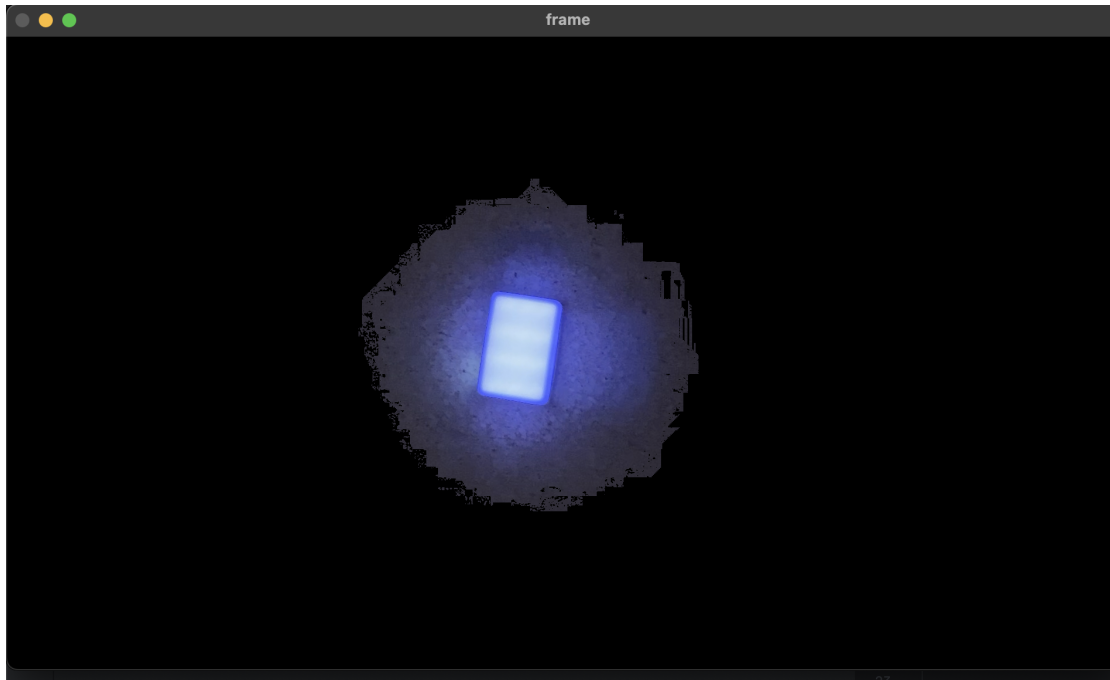


Abbildung 3.7: Ausgabe des Programms zur Erkennung blauer Farbbereiche

Dieses kurze Programm zur Farberkennung wurde mit Hilfe eines Youtube-Tutorials implementiert.[\[39\]](#) Hiervon ausgehend wird dann im nächsten Semester weitergearbeitet. Das Programm wird noch mit den Höhendaten der Drohne zusammengefügt, damit daraus ein Programm zur Kollisionserkennung im dreidimensionalen Raum entsteht. Dieses wird dann noch mit der Drohnensteuerung verbunden, sodass Kollisionen nicht nur erkannt, sondern auch verhindert werden können.

Somit werden wir dann in der Lage sein, unseren Besuchern nicht nur ein beeindruckendes, sondern vor allem auch sicheres Erlebnis zu bieten.

3.7 Externe Komponenten

(Anusha Kanagarasa)

Im Rahmen unseres Projekts nutzen wir verschiedene externe Komponenten, um die Leistung, Sicherheit und Sichtbarkeit unserer Drohnen zu verbessern. Diese externen Komponenten spielen eine entscheidende Rolle bei der Realisierung unserer Drohnenshow und tragen dazu bei, ein beeindruckendes und sicheres Erlebnis für unser Publikum zu schaffen.

Im folgenden Abschnitt werden die wichtigsten externen Komponenten, darunter Heliumluftballons, Sicherheitsnetz und LED-Module, detailliert vorgestellt und ihre Funktionen im Kontext unseres Projekts erläutert.

3.7.1 Heliumluftballons für die Tello-Drohne / Quadcopter

Heliumluftballons dienen als externe Komponenten zur Unterstützung der Tello-Drohne. Bei Verwendung von drei Ballons beträgt die maximale Flugdauer 20 Minuten, bevor sie aufgeladen werden muss - ein Prozess, der etwa 1,5 Stunden dauert. Während der Zwischen-Präsentation verlor die Drohne bei Verwendung von nur 3 Ballons innerhalb von etwa 3 Stunden so viel Helium, dass sie beim Abheben in die Ballons hineinflog. Daher wird empfohlen, 4-5 Ballons zu verwenden, um die Schwebefähigkeit während des gesamten Flugbetriebs zu gewährleisten.



Abbildung 3.8: Tello-Drohne mit Heliumluftballons

3.7.2 Sicherheitsnetz

Ein Sicherheitsnetz ist eine wichtige externe Komponente, die wir einsetzen, um das Publikum zu schützen, falls die Drohne außer Kontrolle geraten sollte. Dieses Netz dient als physische Barriere, um potenzielle Risiken zu minimieren und die Sicherheit während der Veranstaltung zu gewährleisten.



Abbildung 3.9: Sicherheitsnetz

3.7.3 LED Module

Zur Verbesserung der Sichtbarkeit der Drohnen bei Dunkelheit sollen LED Module eingesetzt werden (<https://shop.rc-hangar15.de/VIFLY-STROBE-Anti-Kollisions-LED>). Diese Module werden an den Drohnen angebracht und dienen dazu, ihre Position im Dunkeln deutlich erkennbar zu machen. Dies trägt dazu bei, die Sicherheit zu erhöhen und gleichzeitig ein faszinierendes visuelles Element zu schaffen.



Abbildung 3.10: Anti-Kollisions-LED

3.8 Quadcopter

(Tobias Schwarz)

Neben den Tello Drohnen (Kapitel 3.3) standen uns noch zwei Cheerson CX-10 Mini Quadcopter (im Folgenden kurz *Quadcopter* oder *Drohne* genannt) zur Verfügung. Die grundsätzliche Aufgabenstellung war es, die Quadcopter am Ende via Laptop ansteuern zu können, ein Proof-of-Concept zu erstellen und festzustellen ob die Drohnen weiterverwendet werden können.

Im folgenden werden die Herangehensweise, Probleme, Lösungen und Ergebnisse dieses Semesters – der ersten Hälfte des Projekts – dargestellt und erläutert.

3.8.1 Cheerson CX-100

Bevor weiter auf die Details eingegangen wird, werden zunächst die Quadcopter genauer beschrieben: Die Quadcopter sind kleine 4,5 x 4,5 cm Drohnen, die mit einem Controller bedient werden. Sie waren lange Zeit die kleinsten Drohnen der Welt, wurden jedoch von der Skeye Pico Drohne (2,2 x 2,2 cm) abgelöst. Die Cheerson Drohnen sind in verschiedenen Farben erhältlich, wiegen rund 90 Gramm und kosten – je nach Verkäufer – pro Stück ca. 20€. Die Recherche zeigte, dass es zwei Versionen von ihnen gibt: die einen haben eine rote, die anderen eine blaue Platine, die sich nur im Aufbau und in der Anordnung der einzelnen Komponenten unterscheiden – der Grund hierfür ist bislang nicht bekannt. Auch wenn das zu Beginn nicht wichtig erscheint, so hat sich dieser Unterschied im weiteren Verlauf doch auf einzelne Ideen bzw. Herangehensweisen ausgewirkt (Kapitel 3.8.2). Die Kommunikation zwischen Controller und der Drohne basiert auf dem 2.4GHz Frequenzband und findet erst nach einem manuellen Verbindungsaufbau statt. Falls (wie es bei uns der Fall war) mehrere Cheerson Quadcopter der gleichen Bauart vorliegen, ist es egal welchen Controller man zum Steuern verwendet, da vor jedem Start der Drohne die beiden Parteien erst synchronisiert und verbunden werden müssen. Das geschieht, indem die Drohne nach dem Anschalten zunächst im „Suchmodus“ ist und auf einen Impuls des Controllers wartet. Dieser sendet (durch das Bewegen des Schubhebels in maximal-Stellung und wieder zurück) den Impuls und kann sich somit mit der Drohne verbinden. Hierdurch kann jeder Controller *eine* bestimmte Drohne ansteuern. Wenn die Drohne verbunden ist, hält der Akku für ca. 3-4 Minuten im Dauerbetrieb, bis er wieder geladen werden muss. Während



Abbildung 3.11: Cheerson CX-10 Mini (Drohne und Controller) [6]

des Fluges kann sie im Schweben auch weg driften, dagegen kann man sie Trimmen, also ganz leicht den Schub entgegengesetzt dieser Richtung einstellen. Mit einem Knopfdruck und der Vorgabe der Richtung sind sogar Flips (Überschläge) möglich.

3.8.2 Erste Versuche

Die zwei zur Verfügung stehenden Cheerson CX-10 Minis (orange und blau) wurden getestet, um ein Gefühl für sie zu entwickeln. Dabei ist aufgefallen, dass es nicht so einfach war wie gedacht diese zu steuern, da auch keines der Teammitglieder viel Erfahrung mit Drohnen hatte.

Im ersten Schritt wurde eine der beiden Drohnen und ein Controller in die jeweiligen Einzelteile zerlegt, um den Aufbau dieser besser zu verstehen. Die andere wurde als Rückfalloption zunächst unberührt gelassen. Beim Betrachten der einzelnen Teile ist aufgefallen, dass beide sehr einfach aufgebaut sind. Die Drohne besitzt im Grunde nur die vier Motoren, einen Akku auf der Unterseite. Auf der Platine ist als Hauptbauteil der IC Chip, eine kleine Antenne, ein Gyroskop, um die räumliche Orientierung zu halten und natürlich das 2.4 GHz Funkmodul. Der Controller war auch nicht sehr komplex aufgebaut: die beiden analogen Zwei-Achsen Joysticks, verschiedene Knöpfe zum Trimmen, eine Status-LED, das Batteriefach und auch das 2.4 GHz Funkmodul.



Abbildung 3.12: Drohne zerlegt



Abbildung 3.13: Controller zerlegt

Nun folgten mehrere mögliche Ansätze: Zum einen könnte man die Drohne flashen, also die Firmware löschen und eigenen Code auf die Platine spielen (so wie in einem Blog-Eintrag beschrieben [18]). Das hätte – wie Eingangs erwähnt – mit unserer Version der Platine sogar funktioniert, mit der anderen Art, wäre dies schwieriger geworden. Doch aufgrund der irreversiblen Veränderungen und der dadurch entstehenden Unbrauchbarkeit für kommende Projekte mit diesen Drohnen wurde dieser Ansatz verworfen. Jedoch wurde sich dann für den Versuch, eine Kommunikation via nRF24 Transceiver-Modulen einzurichten, entschieden.

3.8.3 nRF24-Transceiver

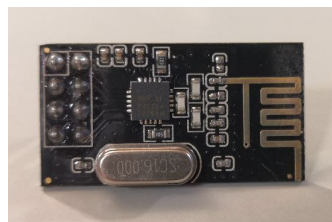


Abbildung 3.14: nRF24 – Transceiver

Neben den Drohnen standen uns mehrere nRF24 Transceiver-Module sowie zwei Arduinos (Mega 2560 und mkr1000) zur Verfügung. Diese Transceiver (**Transmitter** und **Receiver**) sind der inoffizielle Standard für Arduinos und dementsprechend gibt es online auch zahlreiche Dokumentationen und Hilfen. Die Module senden und empfangen auf dem 2.4GHz Frequenzband, haben eine Betriebsspannung von 1,9 – 3,6 V, eine Spannungstoleranz der I/O-Pins von 5V, können auf 125 Kanälen

(2.400 – 2.525 GHz) senden und werden über Arduino und zahlreiche Bibliotheken angesteuert [15].

Zu Beginn wurden diese einzeln und unabhängig von den Drohnen getestet, um die allgemeine Funktionalität zu prüfen. Dabei hat eine online Anleitung zu genau diesen Modulen geholfen [41]. Hier gab es vorgeschriebene Codes für die sog. „Server“ und „Clients“. Die Clients haben aktiv Datenpakete versendet, mit dem Ziel, den Server zu erreichen und eine Antwort zu bekommen. Die Server warten logischerweise auf eine Anfrage und senden daraufhin eine Antwort. Da aber zu Beginn nur ein Teilnehmer bzw. Modul angeschlossen war, konnte niemand antworten. Entweder den Client oder den Server zum Laufen zu bringen war kein Problem, das ging reibungslos: In der Arduino IDE konnte die benötigte Library (RadioHead von Mike McCauley) installiert werden und durch den mitgelieferten Schaltplan in der Anleitung hat das Verkabeln auf dem Breadboard auch schnell funktioniert.

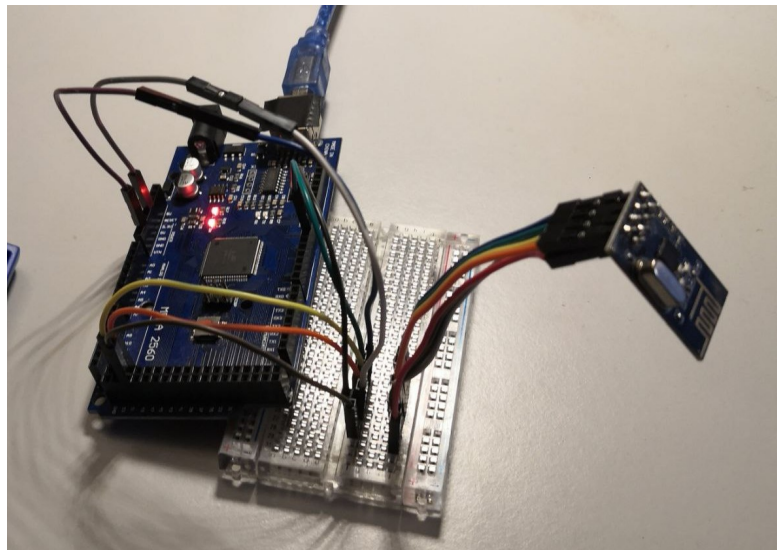


Abbildung 3.15: Aufbau mit Arduino Mega 2560

Listing 3.7: Auszug aus Code für den Client

```
// nrf24_client
#include <SPI.h>
#include <RH_NRF24.h>
// Singleton instance of the radio driver
RH_NRF24 nrf24;

void setup() { [...] }

void loop() {
  Serial.println("Sending to nrf24_server");
  // Send a message to nrf24_server
  uint8_t data[] = "Hello World!";
  nrf24.send(data, sizeof(data));
  nrf24.waitPacketSent();
  // Now wait for a reply
  uint8_t buf[RH_NRF24_MAX_MESSAGE_LEN];
  uint8_t len = sizeof(buf);
  if (nrf24.waitForAvailableTimeout(500)) {
    // Should be a reply message for us now
    if (nrf24.recv(buf, &len)) {
      Serial.print("got reply: ");
      Serial.println((char*)buf);
    }
    else {Serial.println("recv failed");}
  }
  else {Serial.println("No reply, is nrf24_server running?");}
  delay(400);
}
```

Um im nächsten Schritt den Code des Servers zu testen, musste die Verkabelung zum Glück nicht verändert werden und auch der zur Verfügung gestellte Code hat schnell funktioniert.

Nun zum anspruchsvolleren Teil: Der Informationsaustausch der beiden Arduinos via Transceiver miteinander. Zum bisherigen Testaufbau mit dem Mega 2560 und dem nRF24 Modul kam jetzt der mkr1000 dazu. Dieser hat jedoch eine standard-

mäßige Ausgangsspannung von 5V, weshalb zusätzlich noch ein Voltage Divider benötigt wurde, um die 3V Betriebsspannung des Transceivers zu erhalten.

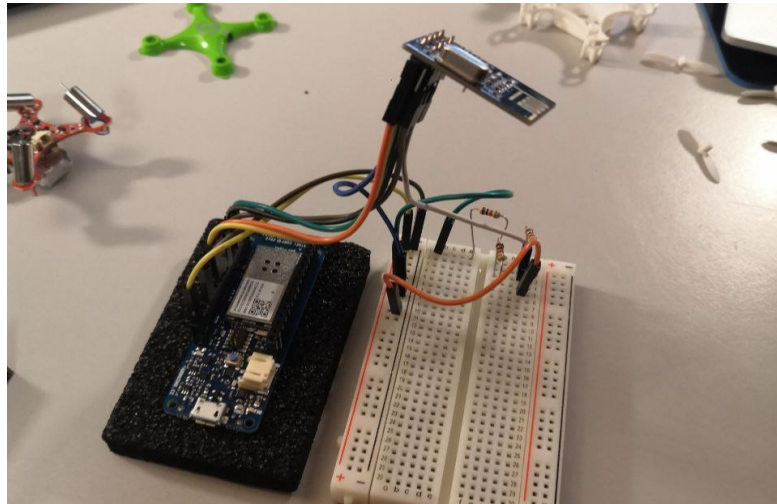


Abbildung 3.16: Aufbau mit Arduino mkr1000 und Voltage Divider

Es kam daraufhin aber zu einigen Problemen, als herausgefunden wurde, dass das Anschließen und separate Betreiben zweier Arduinos am selben Computer nicht vorgesehen ist und auch die COM-Port Zuweisung (welche dem Computer sagt, an welchem Port ein bestimmter Arduino zu finden ist) hier nicht funktionierte. Nach vielen misslungenen Versuchen wurde aufgegeben und ein neuer Ansatz verfolgt, bei welchem die Arduinos an zwei verschiedenen Computern angeschlossen wurden und so miteinander interagieren konnten. Eine weitere Überlegung wäre gewesen, einen Arduino an den Computer anzustecken, den Code hochzuladen, auszuführen und diesen dann mit Akku bzw. externer Stromversorgung zu betreiben. Alles, was der Arduino wissen muss, wurde schon hochgeladen – man kann nur nicht mehr auf den seriellen Monitor zugreifen, um die Terminalausgabe zu sehen. Doch das wäre theoretisch über den zweiten Arduino möglich. Nur wurde aufgrund fehlender Hardware und der erfolgreichen Verwendung von zwei Computern diese Überlegung nicht umgesetzt und wieder verworfen.

Da nun die Kommunikation zwischen den beiden nRF24 Transceivern her- und sichergestellt wurde, sodass diese Daten auf den vorgegeben Frequenzen übertragen können, ist das Ziel – die Drohnen damit anzusteuern – schon näher gekommen. Jetzt war der nächste Schritt überhaupt zu wissen, was der Controller und die Drohnen an Informationen austauschen.

Hier trat aber ein Problem auf. Mithilfe der Transceiver konnten zwar jetzt Daten gesendet und empfangen werden, so wie die Drohne und der Controller es tun, nur gibt es keine Möglichkeit herauszufinden, was genau die beiden Parteien auf den Frequenzen versenden. Selbst wenn man aus den Frequenzen die einzelnen Bits herauslesen könnte; aus den ganzen „1“ und „0“ sind keine verwertbaren Informationen erschließbar. Es gibt online viele Projekte, die das ganze Frequenzband ausgeben können, aber letztendlich wird nur gezeigt, ob auf einer bestimmten Frequenz Daten gesendet werden [14].

Das ist in diesem Fall leider nicht zielführend, da es nicht um die Bits, sondern um den Inhalt der Daten geht. Die letzte Möglichkeit wäre Reverse Engineering, bei dem man versucht, aus bestehenden Systemen Informationen zu extrahieren oder Rückschlüsse auf einzelne Teile zu ziehen. Das hätte jedoch den (zeitlichen) Rahmen gesprengt.

3.8.4 Blimp

Um trotzdem bei den Zwischenpräsentationen etwas vorstellen zu können, musste das Vorgehen angepasst werden – was bedeutete, dass die ganze Arbeit mit den nRF24 Transceiver (zumindest vorerst) verworfen werden musste. Ein neuer Plan war jedoch schnell gefunden. Wir hatten schon vor Beginn des Projekts kurze Infos und Anregungen zu allen Projekte bekommen und mussten uns dann anhand dieser für ein Projekt entscheiden. Prof. Lano hatte uns mehrere Bilder und Videos von Ausstellungen, Künstlern und Tüftlern zur Verfügung gestellt. Eins davon beinhaltete ein YouTube-Video von FliteTest [19], die für ihre ausgefallenen Modellflugbauten bekannt sind. Darin haben sie aus dem Cheerson CX-10 ein Blimp (*engl.* Prallluftschiff) gebaut, sie haben also einen großen Heliumballon genommen und die Drohne etwas modifiziert, darunter gehangen.

Das Ziel ist es, dass der Heliumballons den Auftrieb erzeugt und die Drohne das gesamte Konstrukt lenkt bzw. steuert. Der Vorteil dieses Ansatzes gegenüber dem mit den Transceivern ist, dass hier der original Controller verwendet wird. Das entspricht zwar nicht der Aufgabenstellung "via Laptop steuerbar", wäre aber zumindest ein Proof of Concept. Zum einen für das Thema der interaktiven Drohneninstallation mit (Helium)Ballons und zum anderen für den Hardware-Aspekt, dass dieser zumindest funktioniert und man nun an weiteren Lösungen arbeiten kann.



Abbildung 3.17: Screenshot aus FliteTest Video [19]



Abbildung 3.18: Beim Umlöten der Motoren

Das Blimp besteht, wie schon erwähnt, aus mehreren Heliumballon und der Drohne. Hierbei wird die Drohne auseinandergebaut, die Motoren abgelötet, deren Kabel verlängert und wieder befestigt. Zudem wurde aus sog. Makerfoam (ein dünnes, flaches Material aus Schaumstoff und Papierschichten außen) eine Halterung für die Drohne gebaut. In der Halterung wird die Drohne nun so befestigt, dass die Platine flach auf dem Material liegt. Damit Schub und Höhe steuerbar sind, werden zwei Motoren senkrecht und zwei Motoren waagerecht angebracht.

Um die Idee umzusetzen, musste die Drohne erst auseinandergebaut und die Kabel der Motoren verlängert werden. Hier konnten einfacher Klingeldraht und sehr kleine Schrumpfschläuche verwendet werden. Danach wurde aus Makerfoam ein ähnliches Gebilde wie im FliteTest-Video gebaut. Hierzu wurden auf einem großen Stück des Foams die passenden Teile aufgemalt. Da im Video nicht die genaue Größe der einzelnen Teile angegeben war, wurden eigene Maße verwendet.

Die Halterung ist ca. 15cm lang und 10cm breit. Sie hat grob die Form einer Box ohne Seitenwände und vorne und hinten sind kleine flügelartige Stege, an denen die Rotoren angebracht wurden. Die auseinander gebaute Drohne liegt flach in der Mitte, in der ursprünglichen Orientierung, sodass das Lenken einfacher fällt. Man könnte sie theoretisch beliebig drehen, doch das würde die Steuerung nur unnötig verkomplizieren. Die Kabel der Motoren sind zusammengerollt und in der Haltung befestigt, um nicht in die Propeller zu geraten und so einen Absturz zu verursachen. Der Akku ist mit zwei kurzen Kabeln an der Unterseite der Drohne befestigt und kann dadurch leicht nach hinten weggeklappt werden, wodurch er auf gleicher Höhe mit der Platine und dadurch auch nicht mehr im Weg ist. Die Oberseite der Halterung ist abklappbar, wodurch man einfach an die darin liegende Drohne kommt. Die Stege befinden sich einmal hinten in waagrechter Position, um den Schub zu generieren und einmal an der Front in senkrechter Position, um zusätzlichen Auftrieb zu erzeugen. Auf der Oberseite der Halterung ist eine Befestigung für die Heliumballons angebracht. Diese ist leicht gebogen, um sich besser an die Form der Ballons anzupassen. Je nach Füllgrad der Ballons mit Helium benötigt man 5-6 Stück, um das gesamte Konstrukt ohne Hilfe schweben zu lassen. Es ist jedoch von Vorteil, wenn das ganze Blimp nicht von selbst nach oben schwebt, da es keine elektronische Möglichkeit gibt, es wieder nach unten zu bewegen. Zusätzlich verlieren die Heliumballons sehr schnell das Gas. Diese beiden Punkte kann man zusammengefasst einfach beheben, indem man kleine Gegengewichte auf der Halterung befestigt. Das Blimp muss zu Beginn selbst nach oben fliegen – also zu viel Auftrieb haben – dann werden Gegengewichte befestigt, bis es wieder selbstständig langsam zu Boden geleitet und wenn der Auftrieb der Ballons nachlässt – durch den Gasverlust – bzw. es nicht mehr hoch fliegen kann, kann einfach ein Gegengewicht abgenommen werden.

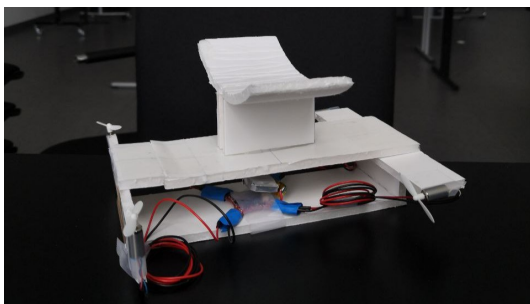


Abbildung 3.19: Blimp seitlich mit Ballonhalterung

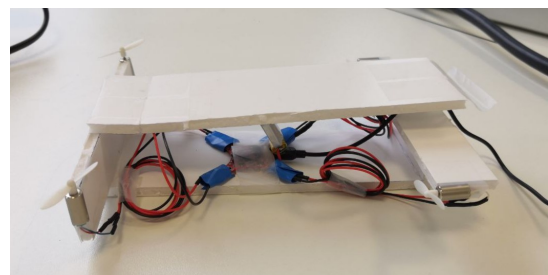


Abbildung 3.20: Blimp seitlich ohne Ballonhalterung

3.8.5 Aktueller Stand



Abbildung 3.21: Blimp bei Zwischenpräsentationen (Insta @media.engineering)

Zusammengefasst wurden die zur Verfügung gestellten Drohnen erforscht und es gab verschiedene Ansätze, in denen versucht wurde, sich in die Steuerung der Drohne zu klinken. Dabei wurde vor allem der Ansatz mit den nRF24 Transceivern und Arduinos verfolgt, bis dieser zunächst aufgegeben werden musste. Im Anschluss wurde, basierend auf einem Video ein kleines Luftschiff mit Heliumballons und der Drohne gebaut. Nach einigem Tüfteln und Probieren ergab das ganze ein Proof of Concept und konnte vorgezeigt werden.

3.8.6 Ziele für nächstes Semester (SoSe24/25)

Für nächstes Semester ist auch schon viel geplant. Es soll ein ESP32 verbaut werden. Das heißt, dass die Platine der Drohne entfernt wird und anstelle davon der ESP eingebaut wird, um die Motoren weiterhin zu betreiben. Dieser ist standardmäßig über WLAN ansteuerbar und schafft damit auch die Möglichkeit der Steuerung über einen Laptop. Hierfür muss der Code dann selbst geschrieben werden. Des Weiteren muss diesbezüglich herausgefunden werden welche Einstellungen sich dafür am besten eignen. Außerdem soll der Akku modular gemacht werden, weil er aktuell nur über eine Kapazität von 100mAh verfügt und diese für eine Flugzeit von nur 3-4 Minuten ausreicht. Es wurden schon andere Akkus bestellt, doch um sie verwenden zu können, muss erst ein passender Anschluss zum Wechseln an die

Drohne gelötet werden. Zusätzlich muss bei den neuen Akkus beachtet werden, dass diese eine größere Kapazität und dadurch ein höheres Gewicht haben. Weil es bei diesem Projekt auf einzelne Gramm ankommt, wird das ein interessanter Punkt im nächsten Semester. Des Weiteren macht es Sinn, die Positionen der Ballons und Rotoren zu verändern um herauszufinden, welche Kombination, Position und Form der Ballons am günstigsten ist. Der letzte Punkt, den ich mir für kommendes Semester vorgenommen habe, ist ein optisch ansprechenderes Gehäuse für die Drohne bzw. den ESP32 zu bauen. Der grundsätzliche Plan ist eben jenes mit einem 3D Drucker zu erstellen, wobei ich hoffe, dass das Material nicht zu schwer ist.

Schlussendlich besteht das nächste Semester aus viel „ausprobieren“ und „tüfteln“, worauf ich schon sehr gespannt bin!

Kapitel 4

Stand der Technik

4.1 Gestenerkennung

(Sina Steinmüller)

Gestenerkennungstechnologie ist heutzutage ein stark erforschtes Themengebiet, das die Computer-Mensch-Interaktion verbessern soll. Aktuell verwendet wird die Gestenerkennung in vielen Bereichen wie beispielsweise der Virtual Reality [48], der Robotersteuerung [30], der Computer oder Tablet Steuerung und vielen anderen Bereichen [51].

Es gibt viele verschiedene Ansätze zur Entwicklung einer guten Gestenerkennung. Einer davon ist der Einsatz von hochwertigem, jedoch kostspieligem Equipment. Hierfür haben sich im Bereich der Kameras überwiegend Tiefenkameras etabliert [29], [35]. Beispiele hierfür sind die Intel RealSense [25] und die Oak-D [32]. Beide der Kameras ermöglichen eine genauere Erfassung von Bewegungen durch Tiefeninformationen. Während die Intel RealSense Kamera vor allem für verschiedene Anwendungen wie Robotik, Augmented Reality und 3-D-Scanning eingesetzt wird, ist die Oak-D Kamera eher für Entwickler von künstlicher Intelligenz und Robotik konzipiert.

Abgesehen vom Einsatz verschiedener Kameramodelle ist auch die Entwicklung neuer, kostengünstigerer Sensoren [33] für die Gestenerkennung ein breit gefächertes Thema. Im Bezug auf die Gestensteuerung der Tellodrohne [24] wird deutlich, dass wir uns bereits ziemlich nah am Stand der Technik befinden, wie ein Paper zur Nutzung von Mediapipe und dem Versuch, 7 Gesten zu implementieren, obwohl das neuronale Netz nur für 6 Gesten ausgelegt ist [34], zeigt. Es gibt hierfür jedoch noch sehr viel Luft nach oben im Bezug auf die genaue Art und Weise der

Implementierung der Gestenerkennung. So ist beispielsweise auch die Erkennung der Region of Interest, abhängig von der Körpermitte, ein möglicher erforschenswerter Ansatz [47].

4.2 Drohnensteuerung

(Maximilian Richter)

Auch in die Drohnensteuerung und die damit einhergehenden Möglichkeiten wird viel Forschungsarbeit investiert. Im Folgenden wird auf aktuelle Erfolge, sowohl im Bereich der Drohnen allgemein, als auch im Bereich von Drohnenschwärmen eingegangen. Außerdem werden die EU-Regelungen zum Umgang mit Drohnen erörtert.

4.2.1 Drohne (allgemein)

Drohnen finden heutzutage weitreichende Einsatzmöglichkeiten. Zum Beispiel in der medizinischen Notversorgung, bei der Drohnen in der Zukunft eine große Rolle bei der Erst-Versorgung von Patienten spielen könnten [27]. Auch für das Auffinden verunglückter Personen wurden bereits funktionierende Konzepte mittels der Verwendung von „Deep CNN Detection“ entwickelt [40].

In der Landwirtschaft spielen Drohnen mittlerweile ebenfalls eine Rolle. Der Weltmarktführer in der Drohnenherstellung „DJI“ hat bereits unter der Modellreihe „DJI Agriculture“ mehrere Drohnen veröffentlicht, die den Ernteschutz und die Pflanzeninspektion vereinfachen und verbessern sollen. Eine höhere Erntequalität, geringere Betriebskosten, sowie eine höhere Ertragsrate sollen ebenfalls möglich sein [42].

Die Verwendung von Drohnen unterliegt in Europa strengen Richtlinien. So benötigt man ab einer Startmasse von 250 Gramm einen Kompetenznachweis, umgangssprachlich auch „Drohnenführerschein“ genannt.

Drohnen werden in drei Betriebskategorien unterteilt [10]:

- „Offen“:
Startmasse < 25kg, innerhalb der Sichtweite, Höhe < 120m

- „Speziell“:
Startmasse > 25kg, außerhalb der Sichtweite
- „Zulassungspflichtig“:
große, schwere Drohnen, z.B. zur Beförderung von Personen/gefährlichen Gütern

Drohnen der Kategorien „Speziell“ und „Zulassungspflichtig“ unterliegen grundsätzlich einer Registrierungspflicht. Drohnen der „offenen“ Kategorie sind registrierungspflichtig ab 250 Gramm, wenn sie mit einem Sensor, der personenbezogene Daten erfassen kann ausgestattet sind (z.B. Kamera).

Genehmigungen müssen für den Betrieb von Drohnen der Kategorie „Speziell“ und „Zulassungspflichtig“ eingeholt werden. „Offene“ Drohnen unterliegen keiner Genehmigungspflicht [10].

4.2.2 Drohnenschwarm

(Lisa Berbig)

Durch den Bedeutungsgewinn von Drohnen in den verschiedenen Bereichen stellt sich auch die Frage, wie es möglich ist, mehrere Drohnen untereinander zu koordinieren.

Ein prominentes Forschungsgebiet zu diesem Thema ist das der ‚Swarm Robotics‘, in dem man sich an Methoden der Bionik bedient. Dafür wird das Verhalten von Schwärmen in der Natur, wie beispielsweise Vogelschwärmen oder Fischeschwärmen, nachgeahmt und auf Roboter, zu welchen auch Drohnen zählen, angewendet.

Dieses Forschungsgebiet hat seinen Ursprung zwar bereits in den 1990er-Jahren, gewann aber erst mit der Jahrtausendwende an Bedeutung. Aktueller Gegenstand der wissenschaftlichen Arbeit ist die Untersuchung der Funktionsweisen und Mechanismen von in der Natur vorkommenden Schwärmen.

Zukünftige Herausforderungen für die Forschung werden unter anderem die Sicherheit im Sinne von Schutz vor Kontrollübernahme der Drohnen durch Cyberangriffe und auch Datenschutz darstellen, aber auch die Implementierung von Machine Learning oder die Interaktion eines Drohnenschwarms mit Menschen.[11]

Auch die Robustheit der Netzwerke, über die Drohnen gesteuert werden, ist Thema der aktuellen Forschung. Besonders beim Einsatz von Drohnen beispielsweise in

Krisengebieten, ist nicht immer eine stabile Netzwerkverbindung gegeben, weshalb hierfür nach verschiedenen Lösungsansätzen gesucht wird.[\[7\]](#)

Das Thema der Steuerung von Drohnenschwärmen ist also ein noch nicht vollständig erforschtes Gebiet, welches auch in Zukunft noch eine bedeutende Rolle spielen wird.

Kapitel 5

Ausblick

Unser Ziel für die Abschlusspräsentation ist es, ein interaktives Erlebnis zu schaffen, das Menschen jeder Altersklasse begeistert. Wir möchten nicht nur inhaltlich, sondern auch optisch mit dem Projekt Aufmerksamkeit auf unsere Idee lenken. Das bedeutet eine perfekt funktionierende Gestensteuerung, hochwertige Werbematerialien und einen Projektraum, der erstaunt und sprachlos macht. Im Laufe des 6. Semesters soll mehr Energie und Arbeit in die Promotion des Projekts gesteckt werden, um weit mehr Reichweite zu generieren und mehr Interesse zu wecken. Dabei sollen nicht nur Mitglieder der Hochschulfamilie, sondern auch Personen in der Gesamtregion Nürnberg, Fürth und Erlangen, große Technikfirmen und Personen aus der Kunstszene erreicht werden. Projektintern werden wir die Kommunikation optimieren, um die Zusammenarbeit zu verbessern. Außerdem arbeiten wir bereits daran, neue Wege zu finden, um mehr Drohnen einsetzen zu können, sodass es auch mehr Einsatzmöglichkeiten für jede einzelne gibt.

5.1 Interaktive Drohnensteuerung

Der aktuelle technische Stand bietet uns verschiedene Möglichkeiten und Ideen, wie es im 6. Semester weitergehen könnte. Bei der Zwischenpräsentation hat das Projekt DRONEVERSE bereits viel Aufmerksamkeit und Interesse sowohl bei den Professoren und Betreuern als auch bei Kommilitonen geweckt. Es gibt 3 mögliche Ideen, die gut umgesetzt werden können, welche im Folgenden beschrieben werden.

5.1.1 Hindernisparcour

Hindernisparcours kennt man bereits von anderen Flugshows, egal ob Flugzeug, Helikopter oder Drohne diese Parcours machen immer sehr viel Spaß. Unsere Überlegung ist es, Ringe auf dem Boden zu platzieren, in denen die Besucher die Drohne landen müssen, im Zickzack zwischen statischen Ballonsäulen fliegen und Extra-Punkte sammeln können, wenn die Drohne durch Lichtschranken fliegt. Mit dieser Strategie können Menschen aller Altersklassen spielerisch an die Drohnen und deren Technik herangeführt werden.

5.1.2 Langzeitbelichtung

Um den künstlerischen Aspekt des Projekts noch einmal aufzugreifen, könnte man an die Traversen an der Decke eine Kamera aufhängen und von den Drohnen ein Foto mit Langzeitbelichtung machen. Dafür werden die Drohnen mit extra LEDs ausgestattet und jede Drohne bekommt eine eigene Farbe zugewiesen. Die Besucher könne Dank der Gestensteuerung eigene Muster mit den Drohnen fliegen und selbst kreativ werden. Von den Bewegungsmustern wird ein Bild angefertigt, welches den Besuchern per Mail zugeschickt wird. Um diese Idee umzusetzen, müssen wir den Raum bestmöglich abdunkeln und die Gestensteuerung so abändern, dass fließende Wisch-Gesten möglich sind.

5.1.3 Drohnenmultiplayergame

Die dritte Idee ist es, ein Multiplayerspiel mit den Drohnen zu entwickeln, denn die Luftballons, welche an den Drohnen angebracht sind erinnern stark an MarioKart

Ballonbattle. Bei MarioKart sind an jedem Kart drei Ballons befestigt und man versucht mit den aufgesammelten Items die Ballons der Gegner zu treffen und dadurch zu gewinnen. Jeder Luftballon symbolisiert ein Leben und es spielen zwei Besucher gegeneinander. Die Spieler müssen versuchen, die Luftballons des Gegners mit einem Laser abzuschießen oder durch eine an der Drohne befestigten Nadel die gegnerischen Ballons zum Platzen zu bringen. Die Option, die Ballons durch einen Laser zum Platzen zu bringen, erfolgt virtuell, sodass zu jeder Zeit für alle beteiligten Personen die Sicherheit gewährleistet ist.

5.2 Raumgestaltung

Für unsere Abschlusspräsentation wollen wir den Raum BB 008 belegen, weil dieser über Deckentraversen verfügt, an denen wir Lichter, Lautsprecher, Netze und unsere Ballons anbringen können. Der Raum ist so geplant, dass man alle möglichen Interaktionen abdecken kann. Das heißt der Hindernisparcours, die Langzeitbelichtung und das Multiplayerspiel sind alle in diesem Raum umsetzbar. Der Raum sollte bestmöglich begehbar sein, sodass wir möglichst viel Platz für unsere Installation und die Besucher haben. Das bedeutet auch alles, was nicht aus dem Raum entfernt werden soll, werden wir hinter Vorhängen verstecken. Wenn man durch die Tür geht, erwartet einen ein Bogenportal aus Luftballons, hinter denen steuerbare pulsierende LEDs sind, sodass man sich fühlt, als würde man ein anderes Universum betreten. Nach dem Eingang soll ein Bildschirm sein, auf dem der Werbetrailer mit dem Kolibri läuft, der nächstes Semester erstellt wird und auch Informationen über unser Projekt zeigt.

In der hinteren Ecke des Raumes sollen die Luftballons statisch installiert werden. Davor soll ein kleiner Informationsstand mit Flyern, Broschüren und weiterem Werbematerial stehen.

Die Mitte des Raumes soll auch hier durch unser Netz abgegrenzt sein, um im Falle eines unkontrollierten Drohnenfluges und eines möglichen Absturzes auf die Besucher stets die Sicherheit gewährleisten zu können. Da die Drohnen selbst noch mit LEDs ausgestattet werden sollen, planen wir die Installation einer Kamera an den Deckentraversen, welche für die mögliche Langzeitbelichtung zuständig ist. Die Besucher sollen durch Gestensteuerung Muster in die Luft fliegen lassen können, während wir davon bleibende Erinnerungsbilder machen. Hinter dem Sicherheitsnetz soll es eine Ladestation für die Drohnenakkus geben, sodass unsere Show ohne größere Pausen kontinuierlich weiterlaufen kann. Vor dem Netz sollen Tische mit Laptops stehen, auf denen die Gestensteuerung der Drohnen läuft. Diese sind als Notlösung geplant, falls eine Gestenerkennung über die Kamera der Drohne nicht möglich ist. Um den Raum auszuleuchten, werden wir einen Kommilitonen fragen, ob er uns seine LED Pars leihen kann. Außerdem möchten wir versuchen, in die statisch befestigten Luftballons LEDs in Plastikkugeln zu integrieren, um die Atmosphäre weiter zu verbessern. Sowohl für die statischen Luftballons als auch für jene die an den Drohnen angebracht sind werden wir hochwertigere Ballons

verwenden. Diese sollen widerstandsfähiger sein, nicht so leicht platzen und auch das Helium soll nicht so schnell entweichen.

Passend zum Thema wird im Raum Musik abgespielt, um auch die Geräusche der Rotoren der Drohnen etwas abzdämpfen und die Gesamtatmosphäre abzurunden. Die Lautstärke wird so eingestellt, dass sich die Besucher noch unterhalten können. Der gesamte Raum soll so angelegt sein, dass die Sicherheit der Besucher gewährleistet ist und niemand im Weg stehen kann. Es soll ein vorgefertigter Weg vorhanden sein, der durch Beleuchtung oder Markierungen am Boden deutlich sichtbar sein soll.

5.2.1 Umsetzung und Durchführung der Präsentation

Für den Aufbau der Zwischenpräsentation haben wir in Raum BB 210 alle Gruppentische entfernt und stattdessen Stühle in der vorderen Hälfte des Raumes aufgestellt. Als Schutzvorrichtung für unsere fliegenden Drohnen haben wir ein Katzennetz verwendet. Dieses Netz besteht aus transparentem Monofilamentgarn, wodurch es leicht durchsichtig ist und den Blick auf die Drohnenshow kaum beeinträchtigt. Vorläufig wurde das Netz mit Paketband am Boden befestigt. Wir arbeiten noch an einer eleganteren Endlösung.

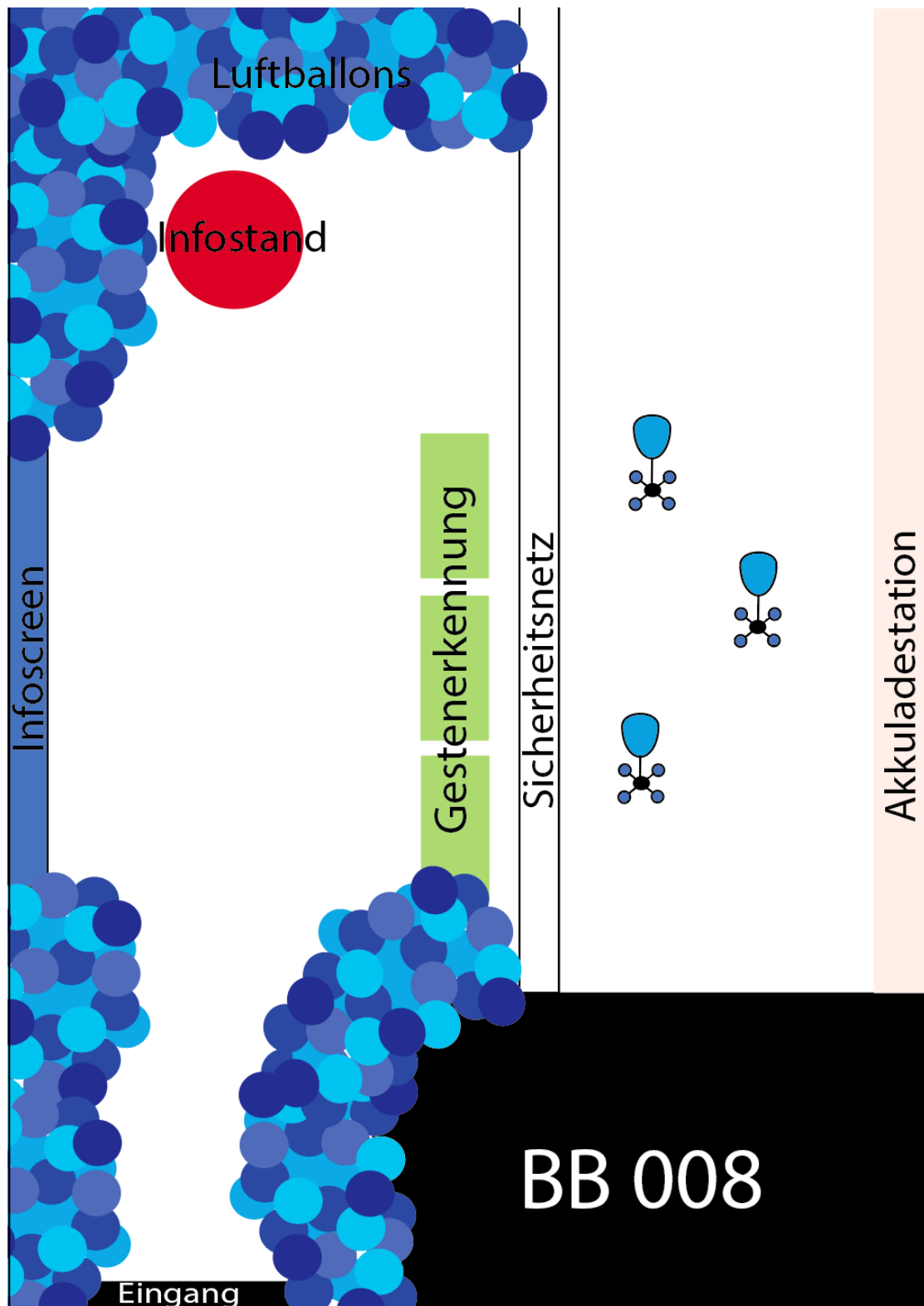


Abbildung 5.1: Raumplanung

Kapitel 6

Zusammenfassung

(Katharina Hölzl)

Zusammenfassend kann man sagen, dass es uns gelungen ist, mit zwei DJI Tello Entwicklungsdrohnen ein interaktives und vorzeigbares Drohnenprojekt zu realisieren. Es ist uns gelungen, eine Gestensteuerung zu implementieren, die jedoch noch optimiert werden muss. Unter anderem dadurch, dass die gestenbasierte Drohnensteuerung noch die Entwicklung genauerer Gesten erfordert, die verwendet werden sollen. Die Entscheidung darüber, wie das Projekt interaktiv gestaltet werden soll, liegt noch vor uns und wird im Kapitel 5.1 betrachtet.

Außerdem ist es uns gelungen, die Drohne erfolgreich auf verschiedene Weisen und mit verschiedenen Programmiersprachen zu steuern. Darüber hinaus gibt es Ansätze für die Schwarmsteuerung, die auf diesem Erfolg aufbauen. Hierbei wird Kollisionsvermeidung ebenso wichtig sein wie präzise Steuerung und Positionierung im Raum. Die Quadcopter sind flugfähig, jedoch muss das Proof of Concept noch überarbeitet werden und die Quadcopter müssen ins Projekt integriert werden. Hier fehlt hauptsächlich noch die Steuerung über den Computer, diese findet momentan noch über eine Fernbedienung statt. Das Corporate Design ist bereits entwickelt. Es fehlen jedoch neue Plakate, Flyer und eine Broschüre, die vor Abschluss des Projekts entworfen werden sollen und alle relevanten Informationen über das Projekt enthalten. Außerdem wird ein Werbetrailer in Blender und eine eigene Webseite erstellt.

Abbildungsverzeichnis

2.1	Unsere Logo-Skizzen	7
2.2	DRONEVERSE-Logo	8
2.3	Corporate Design Farben	11
2.4	Schriftart	13
2.5	Mockup-Notizbuch	14
2.6	Mockup-T-shirt	14
2.7	FlyerFront	15
2.8	FlyerBack	16
2.9	Statistik zu Instagram	19
2.10	Instagram DRONEVERSE	21
2.11	Beitrag DRONEVERSE	22
2.12	Beitrag DRONEVERSE	23
3.1	Screenshot der Obsidian Dokumentation	29
3.2	Screenshot der Trello Aufgabenliste	31
3.3	Handpose Gelenkpunkte	37
3.4	Ausgabe eines Pythonskripts in einer React.js Anwendung	42
3.5	Flowchart Ablauf Drohnensteuerung	52
3.6	Skizze zur Raumaufteilung für die Kollisionserkennung	58
3.7	Ausgabe des Programms zur Erkennung blauer Farbbereiche	61
3.8	Tello-Drohne mit Heliumluftballons	62
3.9	Sicherheitsnetz	63
3.10	Anti-Kollisions-LED	63
3.11	Cheerson CX-10 Mini (Drohne und Controller) [6]	65
3.12	Drohne zerlegt	66
3.13	Controller zerlegt	66
3.14	nRF24 – Transciever	66
3.15	Aufbau mit Arduino Mega 2560	67
3.16	Aufbau mit Arduino mkr1000 und Voltage Divider	69
3.17	Screenshot aus FliteTest Video [19]	71

3.18 Beim Umlöten der Motoren	71
3.19 Blimp seitlich mit Ballonhalterung	72
3.20 Blimp seitlich ohne Ballonhalterung	72
3.21 Blimp bei Zwischenpräsentationen (Insta @media.engineering)	73
5.1 Raumplanung	84

Listings

3.1 Individueller Fingergelenkstyle der Handposegelenke	37
3.2 Rock Geste mit Fingerpose	38
3.3 Pythonintegration in React.js	41
3.4 Modul zur Tastenerkennung - TastenErkennung.py	50
3.5 Modul zur Steuerung der Drohne - DrohnenSteuerung.py	51
3.6 Programm zur Erkennung blauer Farbbereiche in einem Video	60
3.7 Auszug aus Code für den Client	68

Bibliographie

- [1] Sönke Ahrens. *Das Zettelkasten-Prinzip*. Books on Demand, 2017. ISBN: 978-3-7431-2498-1. URL: <https://www.lehmanns.de/shop/sozialwissenschaften/38507220-9783743124981-das-zettelkasten-prinzip>.
- [2] andypotato. *Fingerpose* NPM. 2022. URL: [Fingerpose](#).
- [3] andypotato. *Real-time Hand Gesture Recognition using TensorFlow OpenCV*. 2022. URL: [Fingerpose](#).
- [4] *Ballon Museum Rom Kunstaustellung*. 2022. URL: <https://www.archeoroma.de/veranstaltungen/balloon-museum/>.
- [5] *Ballonetti statische Ballonkunst*. 2024. URL: <https://www.ballonetti.com/skulpturen/>.
- [6] Cheerson. *Cheerson CX10 Mini Drohne - 4CH 2.4G RC Fernbedienung Drohne Quadcopter Hubschrauber LED CX 10 Mini Drone Spielzeug Geschenk für Kinder Geschenk*. URL: <https://rcdrone.top/de/products/cheerson-cx10-mini-drone>.
- [7] Wu Chen u. a. „Toward Robust and Intelligent Drone Swarm: Challenges and Future Directions“. In: *IEEE Network* 34.4 (2020), S. 278–283. DOI: [10.1109/MNET.001.1900521](#).
- [8] Jakob Löw Damià Fuentes Escoté. *DJITelloPy*. 2018. URL: <https://github.com/damiafuentes/DJITelloPy>.
- [9] Jakob Löw Damià Fuentes Escoté. *DJITelloPy Dokumentation*. 2018. URL: <https://djittelopy.readthedocs.io/en/latest/tello/>.
- [10] Bundesamt für Digitales und Verkehr. *EU-Regelungen für Drohnen*. 2021. URL: <https://bmdv.bund.de/SharedDocs/DE/Artikel/LF/drohnen.html>.
- [11] Marco Dorigo, Guy Theraulaz und Vito Trianni. „Swarm Robotics: Past, Present, and Future [Point of View]“. In: *Proceedings of the IEEE* 109.7 (2021), S. 1152–1165. DOI: [10.1109/JPROC.2021.3072740](#).

- [12] René Dudfield. *pygame documentation*. 2023. URL: <https://www.pygame.org/docs/>.
- [13] René Dudfield. *pygame git*. 2023. URL: <https://github.com/pygame/pygame>.
- [14] Earl. *Using an nRF24L01 Module to Scan the 2.4GHz Frequency Range*. 2015. URL: <https://microcontrollerelectronics.com/using-an-nrf24l01-module-to-scan-the-2-4ghz-frequency-range/>.
- [15] Wolfgang Ewald. *nRF24L01 – 2.4 GHz Funkmodule*. 2022. URL: <https://wolles-elektronikkiste.de/nrf24l01-2-4-ghz-funkmodule>.
- [16] Abdur Razzaq Fayjie u. a. „Voice enabled smart drone control“. In: *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*. 2017, S. 119–121. DOI: [10.1109/ICUFN.2017.7993759](https://doi.org/10.1109/ICUFN.2017.7993759).
- [17] *Features von Blender*. 2024. URL: <https://www.blender.org/features/>.
- [18] Felix. *Cheerson cx-10 alternative Firmware*. 2014. URL: <https://www.rcgroups.com/forums/showpost.php?p=30045580&postcount=1>.
- [19] FliteTest. *Make a Blimp from a 16MicroDrone?!*. 2019. URL: <https://www.youtube.com/watch?v=NXb11NqYD-s>.
- [20] Wojciech Giernacki u. a. „DJI Tello Quadrotor as a Platform for Research and Education in Mobile Robotics and Control Engineering“. In: *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2022, S. 735–744. DOI: [10.1109/ICUAS54217.2022.9836168](https://doi.org/10.1109/ICUAS54217.2022.9836168).
- [21] *GIT Repository zur Steuerung mehrerer Drohnen mit dem ESP32*. 2022. URL: <https://github.com/touchgadget/ESPTelloCLI>.
- [22] *Hootsuite Marketing-Experten*. 2024. URL: <https://blog.hootsuite.com>.
- [23] Jiage Huo u. a. „Hand Gesture Recognition with Augmented Reality and Leap Motion Controller“. In: *2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. 2021, S. 1015–1019. DOI: [10.1109/IEEM50564.2021.9672611](https://doi.org/10.1109/IEEM50564.2021.9672611).
- [24] M. Iskandar u. a. „Artificial intelligence-based human gesture tracking control techniques of Tello EDU Quadrotor Drone“. In: *International Conference on Green Energy, Computing and Intelligent Technology (GEN-CITy 2023)*. Bd. 2023. 2023, S. 123–128. DOI: [10.1049/icp.2023.1770](https://doi.org/10.1049/icp.2023.1770).

- [25] Georgi Krastev und Ivan Ralev. „Gesture Based System for User Interface Control“. In: *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. 2021, S. 438–441. DOI: [10.1109/ISMSIT52890.2021.9604693](https://doi.org/10.1109/ISMSIT52890.2021.9604693).
- [26] Neurons Lab. *Drone control via gestures using MediaPipe Hands*. 2021. URL: <https://github.com/kinivi/tello-gesture-control>.
- [27] Kalagani Gopi Lakshmi u. a. „Innovative Usage Of Drones In Medical Emergency“. In: *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*. Bd. 1. 2023, S. 409–414. DOI: [10.1109/ICACCS57279.2023.10112949](https://doi.org/10.1109/ICACCS57279.2023.10112949).
- [28] *Leitfaden zu Instagram-Posts*. 2024. URL: <https://blog.hootsuite.com/de/beste-zeit-fuer-instagram-posts/>.
- [29] David González León u. a. „Video Hand Gestures Recognition Using Depth Camera and Lightweight CNN“. In: *IEEE Sensors Journal* 22.14 (2022), S. 14610–14619. DOI: [10.1109/JSEN.2022.3181518](https://doi.org/10.1109/JSEN.2022.3181518).
- [30] Tiansong Liu. „Research of the Gesture Control System for a Industrial Robot“. In: *2023 3rd International Conference on Electrical Engineering and Mechatronics Technology (ICEEMT)*. 2023, S. 540–543. DOI: [10.1109/ICEEMT59522.2023.10263211](https://doi.org/10.1109/ICEEMT59522.2023.10263211).
- [31] *Lunatx Drohnenshow*. 2024. URL: <https://lunatx.de>.
- [32] Abu Saleh Musa Miah, Md. Al Mehedi Hasan und Jungpil Shin. „Dynamic Hand Gesture Recognition Using Multi-Branch Attention Based Graph and General Deep Learning Model“. In: *IEEE Access* 11 (2023), S. 4703–4716. DOI: [10.1109/ACCESS.2023.3235368](https://doi.org/10.1109/ACCESS.2023.3235368).
- [33] Andrea Mongardi u. a. „Hand Gestures Recognition for Human-Machine Interfaces: A Low-Power Bio-Inspired Armband“. In: *IEEE Transactions on Biomedical Circuits and Systems* 16.6 (2022), S. 1348–1365. DOI: [10.1109/TBCAS.2022.3211424](https://doi.org/10.1109/TBCAS.2022.3211424).
- [34] Jack Nemec und Rocio Alba-Flores. „Unmanned Aerial Vehicle Control Using Hand Gestures and Neural Networks“. In: *2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*. 2022, S. 1–4. DOI: [10.1109/IEMTRONICS55184.2022.9795780](https://doi.org/10.1109/IEMTRONICS55184.2022.9795780).

- [35] Markus Oberweger, Paul Wohlhart und Vincent Lepetit. „Training a Feedback Loop for Hand Pose Estimation“. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, S. 3316–3324. DOI: [10.1109/ICCV.2015.379](https://doi.org/10.1109/ICCV.2015.379).
- [36] Antonello Pastore u. a. „Hierarchical Multi-Robot Fleet Architecture with a Kinematics-adaptive Drive System“. In: *2023 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. 2023, S. 145–150. DOI: [10.1109/ICARSC58346.2023.10129620](https://doi.org/10.1109/ICARSC58346.2023.10129620).
- [37] Nicholas Renotte. *GestureRecognition*. 2020. URL: <https://github.com/nicknochnack/GestureRecognition>.
- [38] Carine Rognon u. a. „FlyJacket: An Upper Body Soft Exoskeleton for Immersive Drone Control“. In: *IEEE Robotics and Automation Letters* 3.3 (2018), S. 2362–2369. DOI: [10.1109/LRA.2018.2810955](https://doi.org/10.1109/LRA.2018.2810955).
- [39] Tim Ruscica. *Tutorial zur Farbwerterkennung mit OpenCV*. 2021. URL: <https://www.youtube.com/watch?v=ddSo8Nb0mTw>.
- [40] Sasa Sambolek und Marina Ivasic-Kos. „Automatic Person Detection in Search and Rescue Operations Using Deep CNN Detectors“. In: *IEEE Access* 9 (2021), S. 37905–37922. DOI: [10.1109/ACCESS.2021.3063681](https://doi.org/10.1109/ACCESS.2021.3063681).
- [41] Rui Santos. *Complete Guide for nRF24L01 – 2.4GHz RF Transceiver Module With Arduino*. 2015. URL: <https://randomnerdtutorials.com/nrf24l01-2-4ghz-rf-transceiver-module-with-arduino/>.
- [42] Da-Jiang Innovations Science und Ltd Technology Co. *DJI Agriculture*. 2024. URL: <https://ag.dji.com/de>.
- [43] K G Shanthi u. a. „Haar Cascade Face Detection and Local Binary Pattern Histogram Face Recognition based Drone“. In: *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*. Bd. 1. 2022, S. 295–298. DOI: [10.1109/ICACCS54159.2022.9785051](https://doi.org/10.1109/ICACCS54159.2022.9785051).
- [44] SKY MAGIC Drone Shows. 2024. URL: <https://skymagic.show>.
- [45] Espressif Systems. *Datenblatt zum ESP32*. 2023. URL: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.
- [46] Ryze Tech. *Tello Technische Daten*. 2023. URL: <https://www.ryzerobotics.com/de/tello>.

- [47] Shoichiro Togo und Hiroyuki Ukida. „Gesture Recognition Using Hand Region Estimation in Robot Manipulation“. In: *2021 60th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. 2021, S. 1122–1127.
- [48] Chia-Chun Tsai, Chih-Chia Kuo und Yen-Lun Chen. „3D Hand Gesture Recognition for Drone Control in Unity“. In: *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. 2020, S. 985–988. DOI: [10.1109/CASE48305.2020.9216807](https://doi.org/10.1109/CASE48305.2020.9216807).
- [49] Dickson Neoh Tze How, Wan Zul Fahmi Bin Wan Ibrahim und Khairul Salleh Mohamed Sahari. „A Dataglove Hardware Design and Real-Time Sign Gesture Interpretation“. In: *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)*. 2018, S. 946–949. DOI: [10.1109/SCIS-ISIS.2018.00157](https://doi.org/10.1109/SCIS-ISIS.2018.00157).
- [50] *Verwendeter ESP32*. URL: <https://www.berrybase.de/esp32-nodemcu-development-board>.
- [51] Rui Wang und Chengjiao Sun. „A Gesture Recognition and Drone Control System based on Residual Neural Network“. In: *2023 IEEE International Conference on Mechatronics and Automation (ICMA)*. 2023, S. 25–30. DOI: [10.1109/ICMA57826.2023.10215930](https://doi.org/10.1109/ICMA57826.2023.10215930).
- [52] Yangguang Yu u. a. „ROS-based UAV control using hand gesture recognition“. In: *2017 29th Chinese Control And Decision Conference (CCDC)*. 2017, S. 6795–6799. DOI: [10.1109/CCDC.2017.7978402](https://doi.org/10.1109/CCDC.2017.7978402).