Prototypische Entwicklung eines Web-basierten 3D-Werkzeugs zur Kamerapositionierung

[Untertitel des Dokuments]

Jan Christmeier

Medieninformatik

[Kurzzusammenfassung 3](#_Toc140788380)

[Einleitung 4](#_Toc140788381)

[Stand der Technik 6](#_Toc140788382)

[Firma IPICA LLC 6](#_Toc140788383)

[Konzept 9](#_Toc140788384)

[Use-Cases 9](#_Toc140788385)

[Aufgabenstellung 12](#_Toc140788386)

[UI 14](#_Toc140788387)

[Vorwissen für die prototypische Umsetzung 14](#_Toc140788388)

[Kameratechnik 14](#_Toc140788389)

[Das 3D Objekt 15](#_Toc140788390)

[Software-Entwicklertools 16](#_Toc140788391)

[Warum Web-Technologie 16](#_Toc140788392)

[Die großen Web Frameworks 16](#_Toc140788393)

[Angular 17](#_Toc140788394)

[Wieso Angular? 18](#_Toc140788395)

[Three.js 19](#_Toc140788396)

[Implementierung 22](#_Toc140788397)

[Angular Implementierung 22](#_Toc140788398)

[Three.js Implementierung 23](#_Toc140788399)

[Kamera Problem 25](#_Toc140788400)

[Komponenten Entwurf 26](#_Toc140788401)

[Evaluierung/Diskussion 26](#_Toc140788402)

[Zusammenfassung 26](#_Toc140788403)

[Ausblick/ Weiterentwicklungsmöglichkeiten 28](#_Toc140788404)

[Verdeckung 29](#_Toc140788405)

[Mathematische Algorithmen 30](#_Toc140788406)

[Weitere Person kann sich durch das 3D Objekt bewegen 31](#_Toc140788407)

[Literaturverzeichnis i](#_Toc140788408)

[Bilder iv](#_Toc140788409)

[Stichpunkte iv](#_Toc140788410)

[Abbildung 1 Import Material-Componente in app-component.ts iv](#_Toc140609591)

[Abbildung 2 Grundlage des Prototypen iv](#_Toc140609592)

# Kurzzusammenfassung

In verschiedenen Bereichen wie Überwachungssystemen und Computergrafik spielt die Betrachtung von Blickpunkten und den Erfassungsbereichen von Kameras eine entscheidende Rolle. In dieser Thesis wird ein System konzipiert und prototypisch implementiert, welches Nutzern die Möglichkeit bietet, Kameras in einem dreidimensionalen Modell, zum Beispiel von Gebäuden, zu positionieren und die Sichtbereiche zu erfassen.

Durch die Möglichkeit, Kameras in einem dreidimensionalen Modell zu platzieren, können Benutzer realitätsnahe und präzise Überwachungsszenarien simulieren. Dies ermöglicht es Sicherheitsfachleuten, optimale Kamerapositionen zu identifizieren, um maximale Abdeckung und Sichtbarkeit zu gewährleisten.

Die Implementierung des Systems beinhaltet die Entwicklung einer benutzerfreundlichen Schnittstelle, die es den Nutzern ermöglicht, intuitiv mit dem dreidimensionalen Modell zu interagieren und die Kameras präzise zu positionieren. Zusätzlich zur Positionierung ermöglicht das System die Festlegung weiterer Parameter wie den Neigungswinkel oder die Brennweite der Kameras, um die Sichtbereiche feiner anzupassen.

# Einleitung

Videoüberwachungssysteme gehören heute zur Standardausstattung vieler Unternehmen und öffentlicher Einrichtungen. Der Einsatz von Kameras hat in den letzten Jahren stark zugenommen und ist zu einem wichtigen Baustein für die Sicherheit von Unternehmen geworden.

Der globale Markt für Videoüberwachungssysteme wurde im Jahr 2020 auf 52,45 Milliarden USD geschätzt und wird voraussichtlich bis 2026 90,37 Milliarden USD erreichen, was einer CAGR von 9,31 % im Zeitraum von 2021 bis 2026 entspricht. (Markt für Videoüberwachungssysteme - Wachstum, Trends, Auswirkungen von Covid-19 und Prognosen (2022 – 2027), 2022)

Rechtliche Rahmenbedingungen:

Bevor Unternehmen in die Videoüberwachung investieren, müssen sie sicherstellen, dass sie die geltenden rechtlichen Bestimmungen einhalten. In Deutschland ist der Einsatz von Überwachungskameras durch das Bundesdatenschutzgesetz und die Datenschutzgrundverordnung geregelt. Das Bundesdatenschutzgesetz besagt, dass die Videoüberwachung nur erlaubt ist, wenn sie zur Wahrung berechtigter Interessen für konkret festgelegte Zwecke erforderlich ist und die Interessen der Betroffenen nicht überwiegen. Die Datenschutzgrundverordnung legt fest, dass Personen über die Videoüberwachung, jeder anderen Art der Datenerhebung und deren Größe und Art informiert werden müssen. Weiterhin muss auch gewährleistet werden, dass die betroffene Person diese Daten jederzeit einsehen oder auch löschen kann. (intersoft consulting services AG, kein Datum)

Sicherheit:

Eine der Hauptgründe für den Einsatz von Videoüberwachungssystemen in Unternehmen ist die Verbesserung der Sicherheit. Mit Kameras können unerwünschte Ereignisse wie Einbrüche, Vandalismus und Diebstahl verhindert oder zumindest aufgezeichnet werden. Allerdings kommt nach Auswertung von 80 Studien in den vergangenen 40 Jahren zu einem anderen Ergebnis:

Videoüberwachung allein ist kein effektives Mittel zur Bekämpfung von Gewaltkriminalität. Studien zeigen, dass ein signifikanter Rückgang von Straftaten nur im Bereich des Drogenhandels und der Eigentumskriminalität zu verzeichnen war. In Bereichen wie Parkplätzen und öffentlichen Verkehrsmitteln führte Videoüberwachung zu einem starken Rückgang von Straftaten, während in Wohngegenden nur ein schwacher Rückgang zu beobachten war und in Innenstädten sowie Wohnhäusern kein signifikanter Rückgang festgestellt wurde. Eine Liveübertragung und aktive Beobachtung sind notwendig, um einen Rückgang von Straftaten durch Überwachungskameras zu erreichen. Zudem sind zusätzliche Maßnahmen wie bessere Beleuchtung, Sicherheitskräfte und Notrufsysteme erforderlich, um die Effektivität der Überwachung zu erhöhen. Studien in Schweden, Kanada und den USA zeigten im Durchschnitt keinen Rückgang von Straftaten durch Überwachungsprojekte, während in Südkorea und Großbritannien signifikante Effekte zu verzeichnen waren. (Piza, Welsh, Farrington, & Thomas, 2019)

Dabei geht es aber hauptsächlich um die Überwachung öffentlicher Räume

Effizienz:

Ein weiterer Grund, warum Unternehmen gerne Videoüberwachungssysteme verwenden, ist die Verbesserung der Effizienz. Mit Kameras können Prozesse und Arbeitsabläufe überwacht und analysiert werden. Dies kann dazu beitragen, Fehler zu reduzieren und die Produktivität zu steigern.

Kostenersparnis:

Videoüberwachungssysteme können auch dazu beitragen, Kosten zu sparen. Sie können dazu beitragen, Verluste durch Diebstahl und Vandalismus zu reduzieren und die Versicherungskosten zu senken. Darüber hinaus können sie dazu beitragen, die Sicherheitskosten insgesamt zu senken, da weniger Sicherheitspersonal benötigt wird.

Akzeptanz:

Obwohl einige Menschen Bedenken hinsichtlich der Privatsphäre haben, zeigen Studien, dass die Mehrheit der Bevölkerung Videoüberwachung in öffentlichen Bereichen und Unternehmen akzeptiert.

Eine Umfrage hat ergeben, dass die breite Bevölkerung in Deutschland auch in Zukunft die Nutzung von Videokameras und Audiolösungen auf Veranstaltungen befürwortet. Die Mehrheit der Befragten (63 Prozent) wünscht sich zusätzliche Kameraüberwachung, um sich in Menschenmengen sicherer zu fühlen. Besonders überzeugt sind die Deutschen von Bodycams, da 73 Prozent der Meinung sind, dass diese das Leben von Polizeibeamten und Sicherheitspersonal, insbesondere bei Großveranstaltungen, erleichtern. Nur eine kleine Minderheit (8 Prozent) glaubt nicht an den Nutzen von Bodycams. Des Weiteren geben 64 Prozent der Umfrageteilnehmer an, dass Kameras ihr Sicherheitsgefühl auf den Straßen erhöhen. Nachts haben fast die Hälfte der Befragten (48 Prozent) Angst, auf deutschen Straßen unterwegs zu sein, und halten aus diesem Grund stets ihr Smartphone griffbereit. Diese Sorge ist auch in Österreich und der Schweiz vorhanden, wenn auch in etwas geringerem Maße. Insbesondere bei den befragten Frauen in Deutschland liegt der Anteil noch höher (über 63 Prozent). (Axis Communications AB, 2021)

Eine andere Studie von SeeTec in Zusammenarbeit mit der Messe "Security Essen" und dem Meinungsforschungsinstitut YouGov zeigt, dass 71 Prozent der Deutschen die Videoüberwachung im öffentlichen Raum befürworten. Besonders hohe Zustimmung gibt es rund um den öffentlichen Nahverkehr, Parkhäuser und Flughäfen. Im privaten Bereich stehen 68 Prozent einer Videoüberwachung positiv gegenüber. Am Arbeitsplatz ist die Akzeptanz gering (9 Prozent). Die Verwendung von biometrischen Daten zur Verbesserung der Sicherheit im öffentlichen Raum wird von 47 Prozent unterstützt. Neue Technologien wie Face-ID haben weniger Zustimmung (43 Prozent). Die Umfrage zeigt, dass sich die Akzeptanz mit der Zeit erhöht. Die Ergebnisse basieren auf einer repräsentativen Befragung von 2.040 Personen. (Riedl, 2018)

# Stand der Technik

Diese Bachelor Arbeit entsteht in Kooperation mit der funkwerk video systeme GmbH, welche auf Videoüberwachung spezialisiert ist. Als Software für die Planung der Lagepläne, wo welche Kamera platziert werden soll, wird über eine von Bosch zur Verfügung gestellten Software gelöst. Diese Software wird von einer kleinen Firma entwickelt.

## Firma IPICA LLC

Die Firma IPICA LLC, welche ihren Hauptsitz in Armenien hat, bietet Software-Tools für die Planung, das Design und die Effizienzanalyse von Videoüberwachungssystemen an (IPICA LLC, 2023):

Diese Software-Tools sind auf die Verbesserung von Sicherheitssystemen spezialisiert. Das Unternehmen bietet eine Reihe von Dienstleistungen an, darunter die Berechnung der besten Kamerapositionen sowie die Bestimmung der Brennweite und des Sichtwinkels der Kameraobjektive. Durch 2D- und 3D-Modellierung können auch die Sichtfelder jeder Kamera geprüft und somit nicht überwachte Bereiche gefunden werden. Die Software ermöglich auch die optische Darstellung von bestimmten Bereichen bzw. Entfernungen, in welchen die Kamera Gesichter oder Kennzeichen erkennen kann.

IPICA LLC ermöglicht auch die Hinzufügung von dreidimensionalen Testobjekten wie Menschen, Autos, Bäumen, Wänden, Fenstern und Türen, um ein realistisches Gebäudemodell zu erstellen. Darüber hinaus kann das Unternehmen auch die erforderliche Bandbreite einschätzen, um Netzwerk-Video-Systeme mit beliebig vielen IP-Kameras und Videoservern zu entwickeln und die erforderliche Größe des Festplattenspeichers für Videoarchive zu schätzen.

Das Unternehmen bietet auch die Möglichkeit, Hintergrundbilder des Lageplans / Raumplans aus AutoCAD DWG(Pro), Visio oder Google Earth zu laden. Kunden können Berechnungen, Zeichnungen und 3D-Modelle in MS Word, Excel, Visio oder anderen Programmen kopieren, um eine Projektdokumentation zu erstellen.

### Fehlende Features

Die von Funkwerk verwendete Software weist eine Fülle an Funktionen auf. Trotzdem sind zwei bedeutende Aspekte nicht in ihrem Funktionsumfang berücksichtigt oder von vornherein nicht geplant gewesen. Hierbei handelt es sich in erster Linie um sich in Echtzeit ändernde Ereignisse sowie physikbasierte Prozesse.

Während die Software zahlreiche Einsatzmöglichkeiten und nützliche Features bietet, zeigt sich ihre Beschränkung, wenn es darum geht, zeitkritische oder dynamische Ereignisse in Echtzeit zu verarbeiten. Situationen, in denen Informationen und Daten sich unmittelbar ändern und eine sofortige Reaktion erforderlich ist.

Darüber hinaus hat die Software keine umfassenden Möglichkeiten für die Simulation von physikbasierten Phänomenen zu bieten. Solche Funktionen wären beispielsweise in Anwendungen, die komplexe physikalische Vorgänge oder realistische Bewegungsabläufe simulieren müssen, äußerst hilfreich.

#### Licht

In Bezug auf die Kamerasoftware ist es wichtig zu beachten, dass diese möglicherweise nicht optimal für Szenarien geeignet ist, in denen eine Szene sowohl sehr helle als auch sehr dunkle Bereiche aufweist. Während das menschliche Auge in der Lage ist, einen breiten Bereich von Helligkeitsstufen wahrzunehmen, stoßen Bildsensoren in Kameras schnell an ihre Grenzen und können unter Überbelichtung oder Unterbelichtung leiden, wodurch wichtige Bilddaten verloren gehen können (IDS Imaging Development Systems GmbH, 2009)

Diese Herausforderung, die als Dynamikumfang bezeichnet wird, kann insbesondere in Außenbereichen mit starkem Lichteinfall auftreten. Wenn die Kamerasoftware, die von Funkwerk verwendet wird, dieses Phänomen nicht angemessen berücksichtigt, könnte dies dazu führen, dass die Kamera Schwierigkeiten hat, alle wichtigen Details in solchen Szenarien einzufangen.

#### Keine Ort änderbare Kamera

Die von Funkwerk verwendete Software bietet in ihrer Planungsfunktionalität lediglich die Möglichkeit, statische Bilder anzuzeigen und Kameras an festen Standorten zu platzieren. Dies bedeutet, dass die Planung von Kameras, die sich auf beweglichen Objekten befinden, nicht unterstützt wird.

Diese Einschränkung kann problematisch sein, insbesondere in Szenarien, in denen die Überwachung von sich bewegenden Objekten erforderlich ist. Beispielsweise können fahrende Roboter oder andere mobile Einheiten, die mit Kameras ausgestattet sind, nicht angemessen in die Planung einbezogen werden.

#### Sound

Es ist wichtig anzumerken, dass die Kamerasoftware keine Soundberechnung in die Planung der Kameras einbezieht. Dies bedeutet, dass die Akustik in der Überwachungsumgebung nicht direkt berücksichtigt wird. Wenn die Klangüberwachung eine wichtige Rolle spielt, beispielsweise um verdächtige Geräusche oder Alarmtöne zu erkennen, kann dies als Nachteil angesehen werden.

In Szenarien, in denen die Soundüberwachung von entscheidender Bedeutung ist, sollten zusätzliche Maßnahmen oder separate Systeme in Betracht gezogen werden, um sicherzustellen, dass sowohl die visuelle als auch die akustische Überwachung gewährleistet sind. Dies könnte beispielsweise die Integration von Mikrofonen oder speziellen Soundüberwachungssystemen umfassen, die unabhängig von der Kamerasoftware arbeiten und eine umfassendere Sicherheitslösung bieten.

Da dies mit der von Funkwerk verwendeten Software nicht möglich ist, müsste bei Bedarf auf eine weitere Software zugegriffen werden.

# Konzept

* Integration in bestehendes System
* Bereits Kenntnisse mit Angular
* Firma noch keine Software für 3D Darstellung
* „Was setzte ich kurz gesagt um“ / Was möchte ich

## Use-Cases

Um die Softwareplanung zu verbessern, wurden drei verschiedene Anwendungsfälle entworfen. Bevor wir jedoch darauf eingehen, ist es wichtig zu verstehen, warum Use-Cases überhaupt sinnvoll sind.

Use Cases sind wertvolle Werkzeuge in der Systementwicklung. Sie ermöglichen eine externe Betrachtung des Systems, indem sie dessen Interaktionen grafisch darstellen. Dadurch gewinnen die Stakeholder ein besseres Verständnis für den Umfang und die Komplexität des Systems. Gut entwickelte Use Cases beschreiben zudem die nach außen sichtbaren Anforderungen des Systems, was es den Stakeholdern ermöglicht, das Design auf seine Zielerfüllung zu überprüfen. Use Cases dienen als Entscheidungshilfe, indem sie eine systematische Bewertung von Optionen und Kompromissen ermöglichen. Sie fördern auch die Kommunikation und Zusammenarbeit zwischen den Stakeholdern, indem sie eine gemeinsame Sprache für das Systemverhalten und die Anforderungen bereitstellen. Durch die Verbindung von Use Cases mit Modellierung und Simulation kann das Systemverständnis weiter verbessert werden, indem verschiedene Szenarien getestet und die Leistung bewertet werden. Dadurch werden fundierte Entscheidungen ermöglicht (Russel, 2019).

### Schulung

Der Use-Case zielt darauf ab, das Wachpersonal eines großen Einkaufszentrums für den Ernstfall zu schulen und die Sicherheitsmaßnahmen zu verbessern. Die interaktive 3D-Objekt-Software spielt dabei eine zentrale Rolle, da sie den Mitarbeitern ermöglicht, das Einkaufszentrum virtuell zu erkunden und in Echtzeit durch das detaillierte 3D-Modell zu navigieren.

Durch die Nutzung der Software können die Mitarbeiter im Voraus die geplanten Positionen der Kameras testen und bewerten. Sie erhalten die Möglichkeit, verschiedene Perspektiven und Blickwinkel der Kameras zu simulieren, um herauszufinden, wie gut sie die kritischen Bereiche abdecken. Dadurch können potenzielle Sicherheitslücken und nicht überwachte Bereiche identifiziert werden.

Während der Schulung lernt das Wachpersonal, wie sie mit den vorhandenen Kameras arbeiten können, um eine umfassende Überwachung der kritischen Bereiche zu gewährleisten. Sie werden darin geschult, die Kamerapositionen zu optimieren, um eine maximale Abdeckung zu erreichen und blinde Flecken zu minimieren. Durch die Interaktion mit der 3D-Objekt-Software können die Mitarbeiter die Auswirkungen ihrer Entscheidungen in Echtzeit beobachten und das Sicherheitssystem des Einkaufszentrums verbessern.

Der Schwerpunkt liegt darauf, dass das Wachpersonal ein fundiertes Verständnis dafür entwickelt, wie die Kameras eingesetzt werden können, um die Sicherheit zu maximieren. Sie lernen, potenzielle Sicherheitslücken zu erkennen und zu schließen, indem sie die Kameras an strategisch wichtigen Stellen positionieren und ihre Blickwinkel optimieren. Gleichzeitig wird ihnen gezeigt, wie sie eine umfassende Überwachung der kritischen Bereiche gewährleisten können, um verdächtige Aktivitäten frühzeitig zu erkennen und schnell reagieren zu können.

Durch die Schulung mit der interaktiven 3D-Objekt-Software kann das Wachpersonal seine Fähigkeiten verbessern und das Einkaufszentrum effektiver schützen. Die virtuelle Erkundung ermöglicht es ihnen, sich in einer realistischen Umgebung mit den Sicherheitsmaßnahmen vertraut zu machen und ihre Entscheidungsfindung zu schärfen.

### Bewegende Kameras

Im vorliegenden Use-Case wird erläutert, wie ein fahrender Roboter oder eine fahrende Kamera innerhalb eines 3D-Objekts platziert und konfiguriert wird. Im Folgenden werden beide Möglichkeiten nur noch als Roboter definiert. Die Software ermöglicht es dem Benutzer, die Perspektive des Roboters in einer virtuellen Umgebung zu erleben. Der Benutzer hat die Möglichkeit, den Roboter eigenständig zu steuern und ihn durch die Umgebung zu bewegen. Während der Simulation kann der Benutzer die virtuelle Sicht des Roboters in Echtzeit überwachen und detailliert prüfen, um mögliche Sichtbehinderungen oder Einschränkungen zu erkennen und das Sicherheitskonzept zu überprüfen.

Der Ablauf des Use-Cases beginnt damit, dass der Benutzer den fahrenden Roboter an einer gewünschten Startposition innerhalb des 3D-Objekts platziert. Dabei hat der Benutzer die Möglichkeit, relevante Eigenschaften wie Größe, Geschwindigkeit und die Brennweite der Kamera entsprechend zu konfigurieren, um eine präzise Anpassung an die Umgebung und den Einsatzzweck zu gewährleisten.

Nachdem der Roboter platziert und konfiguriert wurden, stellt die Software die Sicht des Roboters virtuell dar. Durch diese Simulationsansicht kann der Benutzer das Umfeld aus der Perspektive des Roboters betrachten und somit das Verhalten in der Umgebung besser einschätzen.

Sobald der Roboter einsatzbereit ist, kann der Benutzer sie eigenständig steuern und durch die virtuelle Umgebung bewegen. Während der Simulation hat der Benutzer die Möglichkeit, die virtuelle Sicht in Echtzeit zu überwachen. Dabei werden mögliche Sichtbehinderungen oder Einschränkungen identifiziert und bewertet, um das Sicherheitskonzept zu optimieren.

Die Fähigkeit, den Roboter oder die Kamera in einer sicheren virtuellen Umgebung zu testen, ermöglicht es dem Benutzer, potenzielle Probleme und Risiken frühzeitig zu erkennen. Basierend auf den Erkenntnissen aus der Simulation können bei Bedarf Anpassungen am Sicherheitskonzept oder der Umgebung vorgenommen werden, um die Sicht und Fahrweise des Roboters zu optimieren und potenzielle Probleme zu beheben.

### Echtzeit Verfolgung

In diesem Use-Case wird eine kontinuierliche visuelle Überwachung eines bestimmten Objekts oder einer Person ermöglicht. Durch den Einsatz der Kamerasoftware wird die automatische Verfolgungsfunktion aktiviert, die es den Kameras ermöglicht, das Zielobjekt oder die Zielperson stets im Blickfeld zu behalten.

Der Benutzer beginnt damit, das gewünschte Objekt oder die gewünschte Person auszuwählen, die von den Kameras verfolgt werden sollen. Anschließend aktiviert der Benutzer die Verfolgungsfunktion in der Kamerasoftware und gibt an, welches Zielobjekt oder welche Zielperson verfolgt werden soll. Die Kameras passen ihre Ausrichtung, Schwenk- und Neigebewegungen kontinuierlich an, um sicherzustellen, dass das Zielobjekt oder die Zielperson immer im Blickfeld bleibt. Die Kamerasoftware analysiert die Bilddaten in Echtzeit und aktualisiert die Kamerapositionen entsprechend, um eine präzise Verfolgung sicherzustellen.

Ein praktisches Anwendungsbeispiel für diese Funktion könnte in einem Lagerhausumfeld liegen. Die Kameras werden strategisch im Lagerhaus positioniert und auf die Gabelstapler ausgerichtet. Jeder Gabelstapler ist mit einem Identifikationsmerkmal ausgestattet, das von der Kamerasoftware erkannt werden kann. Sobald ein Gabelstapler erkannt wird, aktiviert die Kamerasoftware automatisch die Verfolgungsfunktion für diesen bestimmten Gabelstapler.

Während des normalen Lagerbetriebs bewegen sich die Gabelstapler kontinuierlich durch das Lagerhaus, um Waren zu transportieren. Die Kamerasoftware verfolgt automatisch die Bewegungen der Gabelstapler und passt die Kamerapositionen an, um sie immer im Blickfeld zu behalten. Auf diese Weise können mögliche Bereiche identifiziert werden, an denen der Gabelstapler hinter anderen Objekten verschwinden kann, was zu unüberwachten Stellen führen könnte.

## Aufgabenstellung

Im Rahmen dieser Entwicklungsarbeit wird eine Softwarelösung prototypisch umgesetzt, die es ermöglicht, dreidimensionale Objekte nahtlos in eine Web-Oberfläche zu integrieren und eine interaktive Navigation durch sie zu gewährleisten. Ein Hauptaugenmerk liegt dabei auf der vielfältigen Darstellung des 3D-Objekts aus verschiedenen Kamerawinkeln. Die Software bietet dem Benutzer die Flexibilität, das Objekt von verschiedenen Blickwinkeln aus zu betrachten, indem sie mehrere Kameras virtuell positioniert. Der Benutzer kann zusätzliche Kameras erzeugen und bei Bedarf auch Kameras wieder entfernen. Während der Erkundung des Objekts wird der Benutzer von bereits platzierten Kameras in der virtuellen Welt beobachtet und in den entsprechenden Kameraperspektiven angezeigt, wenn er sich im Sichtfeld befindet. Die Kameras sollen den Benutzer auch verfolgen können, wenn die Kamera entsprechend eingestellt wurde.

Die Software stellt dem Benutzer zwei verschiedene Fortbewegungsmethoden in der virtuellen Welt zur Verfügung. In der ersten Methode kann sich der Benutzer ähnlich wie ein gewöhnlicher Mensch bewegen und bleibt dabei stets mit dem virtuellen Boden verbunden. In diesem Modus kann der Benutzer sich durch das Objekt bewegen, springen und die Umgebung erkunden. Diese Bewegungsart ermöglicht es dem Benutzer, die räumlichen Beziehungen innerhalb des 3D-Objekts zu erfassen und sich anhand dieser Informationen in der virtuellen Welt zu orientieren.

Um die Bewegung in einem virtuellen Objekt mittels dieser Bewegungsmethode zu ermöglichen, ist eine Kollisionserkennung zwischen dem Benutzer und dem Objekt erforderlich. Kollisionen spielen eine entscheidende Rolle, um eine realistische Bewegung zu ermöglichen und dem Benutzer die Interaktion mit der virtuellen Umgebung zu ermöglichen. Durch Kollisionserkennung wird festgestellt, ob und wie Objekte aufeinandertreffen, und entsprechende Reaktionen wie Ereignisauslösungen oder Soundeffekte werden ausgelöst. Darüber hinaus sind Kollisionen in 3D-Anwendungen wie Videospielen ein wichtiger Bestandteil der Physiksimulation, da sie sicherstellen, dass die Interaktion zwischen den Objekten den physikalischen Gesetzen folgt (Gregory, 2018). Nur somit wird eine realistische Bewegung des vom Spieler gesteuerten Charakters durch das Objekt ermöglicht.

Die zweite Bewegungsmethode, bekannt als "Drohnenansicht", hebt die Einschränkungen der simulierten Gravitation auf und ermöglicht es dem Benutzer, frei und ohne jegliche Limitierungen in der virtuellen Welt zu fliegen. In diesem Modus kann der Benutzer mühelos Punkte identifizieren, an denen zusätzliche Kameras platziert werden können. Diese Bewegungsmethode eignet sich besonders gut für das gezielte Platzieren von Kameras an strategisch wichtigen Positionen.

Diese Software wird unter Verwendung von Webtechnologien entwickelt, um sicherzustellen, dass sie nicht nur auf einer Vielzahl von Geräten verwendet werden kann, sondern auch nahtlos in bereits bestehende Webanwendungen von Funkwerk integriert werden kann. Durch die Nutzung von Webtechnologien wird eine hohe Kompatibilität und Flexibilität gewährleistet, da sie auf verschiedenen Plattformen und Betriebssystemen funktioniert. Dies ermöglicht es Funkwerk´s Mitarbeitern, die Software auf ihren bevorzugten Geräten und über ihre vertrauten Webanwendungen zu nutzen, ohne zusätzliche Installationen oder Anpassungen vornehmen zu müssen. Dieser Vorteil wäre auch für die Kunden zugänglich, wenn Diese Zugriff auf die Software erhalten. Die nahtlose Integration der Software in die vorhandenen Webanwendungen erleichtert auch die Arbeit mit den bereits etablierten Tools und Systemen von Funkwerk, was zu einer effizienteren und effektiveren Nutzung führt.

UI  
Um eine größere Anzahl von gleichzeitig sichtbaren Kamerafenstern darzustellen, bietet sich eine Gitteranordnung an, ähnlich wie bei Monitorwänden. Diese Art der Anordnung wird auch oft in Überwachungssystemen verwendet. Der Vorteil eines solchen Gitters liegt darin, dass es sich flexibel anpassen kann, ohne dass die Größe der einzelnen Kamerastreams begrenzt werden muss. Dadurch können alle Kameras gleichzeitig angezeigt werden, unabhängig von ihrer Anzahl. Natürlich müssen die einzelnen Kamerastreams auf dem Monitor immer kleiner werden, je mehr solcher angezeigt werden. Nur so kann sichergestellt werden, dass auch alle Streams gleichzeitig angezeigt werden können.

Da es sich bei der Software um eine virtuelle Anwendung handelt, ist es möglich, die Kamerastreams in großen Fenstern anzuzeigen, wenn nur eine kleine Anzahl von Kameras vorhanden ist. Wenn jedoch mehr Kamerastreams hinzukommen, wird die Anzahl der Spalten und Reihen im Gitter dynamisch angepasst, um genügend Platz für alle Streams zu schaffen. Somit wird kein Platz ungenutzt gelassen.

Um die Kameras und ihre Parameter einzustellen, ist es wichtig, eine verständliche und übersichtliche Schnittstelle bereitzustellen. Sobald eine Kamera erstellt wird, müssen entsprechende Einträge in den Einstellungen erzeugt und angezeigt werden. Dadurch kann der Benutzer auf einfache Weise die Einstellungen für jede einzelne Kamera anpassen. Gleichzeitig sollte es auch eine Möglichkeit geben, erstellte Kameras zu löschen, was ebenfalls die entsprechenden Einträge in den Einstellungen entfernt.

# Vorwissen für die prototypische Umsetzung

## Kameratechnik

Das Modell der Kamera, das für das Projekt verwendet wird, ist stark vereinfacht und berücksichtigt keine komplexen optischen Konstruktionen, die in einer realen Kamera vorhanden sein können. Im Wesentlichen bedeutet dies, dass die Kameras im Projekt keine Verzerrungen, optischen Fehler oder Beugungsphänomene aufweisen.

Die Kamera im Projekt liefert immer eine perfekte Abbildungsleistung und es gibt keine Begrenzungen in Bezug auf die Menge des einfallenden Lichts oder die Schärfentiefe. Mit anderen Worten, die Kamera kann jedes Objekt unabhängig von seiner Entfernung oder Größe perfekt abbilden, ohne dass es zu Unschärfen, Rauschen oder Verzerrungen kommt.

Aufgrund dieser starken Vereinfachung ist nur ein Parameter der Kamera von entscheidender Bedeutung.; Die Brennweite. Mit dieser kann der anzuzeigende Bildausschnitt gewählt werden.

## Das 3D Objekt

Der Prototyp verwendet 3D Objekte des Datentyps glTF, um diese darstellen zu können. Das glTF-Spezifikationsdokument beschreibt das glTF-Dateiformat, das eine effiziente und erweiterbare Methode für die Übertragung und das Laden von 3D-Inhalten bietet. Es dient als Schnittstelle zwischen 3D-Content-Erstellungswerkzeugen und modernen Grafikanwendungen. Die Motivation und Designziele von glTF werden erläutert. Es soll ein kompaktes und effizientes Format sein, das plattformunabhängig und für verschiedene 3D-Grafik-APIs geeignet ist. glTF zielt auf Laufzeit-Effizienz ab und bietet eine gemeinsame und interoperable Darstellung von 3D-Assets für Endbenutzer. Das Dateiformat hat ebenfalls das Ziel, kompakte Dateigrößen zu haben und auf verschiedenen Geräten und Plattformen einsetzbar zu sein. Es ermöglicht die Darstellung vollständiger 3D-Szenen und ist erweiterbar, um zukünftige Entwicklungen zu berücksichtigen. (International Organization for Standardization & International Electrotechnical Commission, 2022)

Ein weiterer großer Vorteil von glTF Objekten ist, das *.blend* Files, welche mit Blender geöffnet werden können, in dieses Dateiformat exportiert werden können. Blender ist eine kostenlose und Open-Source-Software zur 3D-Erstellung. Es unterstützt den gesamten 3D-Prozess, einschließlich Modellierung, Rigging, Animation, Simulation, Rendering, Compositing und Bewegungsverfolgung, sowie sogar Video-Bearbeitung und die Erstellung von Spielen. Sie darf auch für alle Anwendungen, egal ob kommerziell oder nicht, verwendet werden (Blender Foundation, 2023).

# Software-Entwicklertools

## Warum Web-Technologie

* Was heb Web von anderen ab

## Die großen Web Frameworks

Die zwei großen Alternativen zu Angular sind React und Vue.js. Seit Ende 2016 gibt es auch noch ein weiteres Framework namens Svelte (Harris, 2016), welches stark an Popularität gewinnt. Allerdings ist dies noch nicht so verbreitet, wie die drei großen Javascript basierten Web-Frameworks. Eine jährlich erscheinende ausführliche Umfrage mit im Jahr 2022 mit 73.268 Software-Entwicklern ergab zusammengefasst folgende Trends bei den Web-Frameworks (Stack Overflow, 2022):

In Bezug auf die Popularität liegt React.js mit einem Anteil von 42.62% deutlich vorne und ist damit das am häufigsten verwendete Framework bzw. die am häufigsten verwendete Bibliothek. Es folgen Angular mit einem Anteil von 20.39% und Vue.js mit einem Anteil von 18.82%. Angular.js hat einen Anteil von 8.99% und Svelte liegt mit einem Anteil von 4.58% am unteren Ende der Skala.

Die Zufriedenheit der Entwickler mit diesen Frameworks und Bibliotheken variiert erheblich. Svelte ist derzeit das am meisten geschätzte Framework mit einer Zufriedenheitsrate von 75.28%, gefolgt von React.js mit 68.19% und Vue.js mit 63.16%. Angular hat einen Zufriedenheitswert von 52.27%, während Angular.js mit einem Wert von 21.01% am wenigsten zufriedenstellend ist.

Hinsichtlich der Wünsche von Entwicklern zeigt sich, dass React.js mit 22.54% am meisten gewünscht ist, gefolgt von Vue.js mit 14.6%. Angular und Svelte haben einen Wunschanteil von 7.18% bzw. 9.34%, während Angular.js mit 4.32% am wenigsten gewünscht ist.

Insgesamt zeigen diese Zahlen, dass React.js derzeit die beliebteste Bibliothek oder das beliebteste Framework unter Entwicklern ist und auch die meisten Wünsche in der Entwicklergemeinschaft weckt. Svelte wird von Entwicklern am meisten geschätzt, während Angular am wenigsten zufriedenstellend ist und auch am wenigsten gewünscht wird. Dennoch gehört es zu den am meisten Verwendeten.

## Angular

Angular ist ein weit verbreitetes Framework für die Entwicklung von Frontend-Webanwendungen. Es eignet sich besonders gut für die Erstellung von Single-Page-Anwendungen und dynamischen Webanwendungen. Mit einer Vielzahl von Funktionen und Werkzeugen ermöglicht es eine schnelle und vereinfachte Entwicklung von Webanwendungen. Es ist jedoch wichtig, die Vor- und Nachteile von Angular zu berücksichtigen, um fundierte Entscheidungen zu treffen: (DDI Development, 2019) (Оглукян, 2020)

Vorteile:

* Komponentenbasierte Architektur: Angular basiert auf einer komponentenbasierten Architektur, die es Entwicklern ermöglicht, große und komplexe Anwendungen in kleinere, wiederverwendbare Komponenten zu unterteilen. Dadurch wird die Entwicklung und Wartung von Anwendungen vereinfacht und beschleunigt.
* Two-Way-Data-Binding: Angular verwendet Two-Way-Data-Binding, ein Mechanismus, der Änderungen an der Benutzeroberfläche automatisch an das zugrunde liegende Datenmodell überträgt und umgekehrt. Dadurch wird die Entwicklung von anspruchsvollen Anwendungen vereinfacht und beschleunigt.
* Dependency Injection: Angular verwendet Dependency Injection, ein Entwurfsmuster, das die Abhängigkeiten zwischen den Komponenten verwaltet. Dadurch wird die Entwicklung und Wartung von Anwendungen vereinfacht und beschleunigt.
* Routing: Angular verfügt über ein leistungsstarkes Routing-System, das das Navigieren zwischen verschiedenen Ansichten innerhalb der Anwendung erleichtert. Es ermöglicht auch das dynamische Laden von Code, wodurch die Anwendung schneller und effizienter wird.
* Forms-Modul: Angular bietet ein leistungsstarkes Forms-Modul, das das Erstellen von Formularen und die Validierung von Benutzereingaben erleichtert. Es bietet eine Vielzahl von Funktionen, die die Entwicklung von Formularen vereinfachen und beschleunigen.

Nachteile:

* Komplexität: Angular ist ein sehr leistungsfähiges Framework, aber auch sehr komplex. Es erfordert eine steile Lernkurve und erfordert viel Zeit und Mühe, um sich in das Framework einzuarbeiten.
* Größe: Angular ist auch ein großes Framework, das viel Code benötigt, um ausgeführt zu werden. Das bedeutet, dass Angular-Anwendungen im Vergleich zu anderen Frameworks größer sein können, was zu längeren Ladezeiten führen kann.
* Performance: Angular-Anwendungen können manchmal langsamer sein als andere Frameworks, da sie eine Menge Code benötigen, um ausgeführt zu werden. Allerdings hat Angular in den letzten Jahren große Fortschritte in Bezug auf die Performance gemacht.
* Kompatibilität: Angular ist nicht vollständig kompatibel mit anderen Frameworks und Bibliotheken. Wenn Sie bereits eine Anwendung haben, die auf einem anderen Framework oder einer anderen Bibliothek basiert, kann es schwierig sein, sie auf Angular zu portieren.

Insgesamt hat Angular viele Vorteile, aber auch einige Nachteile, die bei der Wahl des richtigen Frameworks für die Entwicklung von Webanwendungen berücksichtigt werden sollten.

## Wieso Angular?

Die Entscheidung, Angular zu verwenden, basiert oft auf der Tatsache, dass es bereits in der eigenen Firma im Einsatz ist. Diese bereits vorhandene Nutzung von Angular bietet den Vorteil, dass es einfach in bestehende Projekte integriert werden kann.

Ein weiterer wichtiger Grund für die Verwendung von Angular ist die Tatsache, dass es trotz der verhältnismäßig vielen bereits erwähnten negativen Punkten, eine weit verbreitete Plattform ist. Dies bedeutet, dass es eine große Anzahl von Ressourcen, Community-Support und Entwickler-Know-how gibt, die bei der Implementierung und Wartung von Angular-Anwendungen hilfreich sind. Entwickler können auf eine Fülle von Tutorials, Dokumentationen, Foren und anderen Online-Ressourcen zurückgreifen, um Probleme zu lösen oder Fragen zu klären. Die reichhaltige Community rund um Angular sorgt dafür, dass auch Entwickler-Einsteiger viel Hilfe erwarten können.

Die Nutzung einer vertrauten Technologie wie Angular bietet auch den Vorteil einer erhöhten Entwicklungseffizienz. Da Angular bereits in der Firma verwendet wird, entfällt die Notwendigkeit, sich in ein neues Framework einzuarbeiten. Die Entwickler und auch ich sind bereits mit den Grundlagen von Angular vertraut und können daher schneller produktiv arbeiten.

Weiterhin kann Angular-Material verwendet werden. Es stellt eine Sammlung von vorgefertigten, reaktiven und vor-gestalteten UI-Komponenten bereit, die in Angular-Anwendungen verwendet werden können, um eine einheitliche Benutzeroberfläche zu erstellen. Angular Material bietet eine breite Palette an UI-Komponenten wie Schaltflächen, Formularfelder, Dropdown-Listen, Karten, Dialoge, Tabellen, Tooltips und viele weitere. Durch die Verwendung von Angular Material kann Zeit gespart werden, da nicht von Grund auf eigene UI-Komponenten erstellt werden müssen. Stattdessen können die vorgefertigten Komponenten von Angular Material genutzt werden und in die Anwendung integriert werden. Diese Komponenten sind gut dokumentiert und bieten eine Vielzahl an vorgefertigten Beispielen. (Google LLC, 2023)

Ein weiterer wichtiger Aspekt ist, dass der Hauptbestandteil des Codes Three.js ist. Da Angular ein flexibles und erweiterbares Framework ist, kann es problemlos mit anderen Bibliotheken und Frameworks wie Three.js erweitert werden. Daher hat die Wahl des Frameworks keinen großen Einfluss auf die Verwendung von Three.js als Hauptbestandteil des Codes. Angular und Angular-Material dienen hauptsächlich der Strukturierung der Seite, während Three.js für nahezu die gesamte Funktionalität der Anwendung verantwortlich ist.

## Three.js

Three.js ist eine JavaScript-Bibliothek, die es Entwicklern ermöglicht, 3D-Grafiken und -Animationen in Webanwendungen zu erstellen. Es ist eine Open-Source-Bibliothek, die auf WebGL basiert und eine Reihe von Funktionen und Tools für die Erstellung von 3D-Modellen, Texturen, Materialien und Beleuchtungen bereitstellt. Mit Three.js können Entwickler interaktive 3D-Visualisierungen und -Simulationen erstellen, die in Webbrowsern und mobilen Geräten ausgeführt werden können. Es ist eine beliebte Wahl für die Entwicklung von 3D-Spielen, virtuellen Welten, 3D-Diagrammen und Datenvisualisierungen. (Fundamentals, kein Datum) (Responsive Design, kein Datum)

Three.js bietet die Möglichkeit, 3D-Objekte in eine Weboberfläche zu laden, eine Navigation durch diese Objekte zu ermöglichen, die Sicht einer Kamera in dem Objekt anzuzeigen und gezieltes Platzieren von Objekten zu ermöglichen. Diese essenziellen Teilbereiche des Projekts werden jeweils in eigenen Beispielen von Three.js demonstriert. Diese und noch deutlich mehr andere Beispiele können auf der Three.js examples Homepage besichtigt und ausprobiert werden (examples, 2023).

Für das Laden von 3D-Objekten wird ein Beispiel von Three.js verwendet, das zeigt, wie ein bestimmtes Dateiformat, wie zum Beispiel das glTF-Format, in die Szene geladen werden kann. Dabei werden alle erforderlichen Schritte erklärt, um das 3D-Objekt korrekt darzustellen und mit ihm zu interagieren.

Die Navigation durch die 3D-Szene wird ebenfalls anhand eines separaten Beispiels von Three.js veranschaulicht. Hierbei wird gezeigt, wie eine Kamera in der Szene platziert und gesteuert werden kann, um verschiedene Perspektiven und Blickwinkel auf das 3D-Objekt zu erhalten. Dabei können Benutzer beispielsweise die Kamera drehen, zoomen oder bewegen, um das Objekt aus verschiedenen Blickwinkeln zu betrachten.

Ein weiteres Beispiel von Three.js konzentriert sich auf das Platzieren von Objekten in der Szene. Hierbei wird gezeigt, wie 3D-Objekte an bestimmten Positionen platziert werden können, um beispielsweise eine Szene oder Umgebung zu gestalten. Dies ermöglicht es den Benutzern, ihre eigenen 3D-Welten zu erstellen und Objekte nach ihren individuellen Vorstellungen zu positionieren.

Diese Bibliothek wurde vom User *mrdoob* über die Plattform Github zur freien Verfügung gestellt. Einzelne Bestandteile dieser Bibliothek sind Eigentum verschiedener Github-Benutzer (Herzog, 2021). Laut Github verwenden über 160.000 Accounts diese Bibliothek für ihre eigenen Projekte (GitHub, Inc., 2023).

### Beispiele

Auf der Startseite von Three.js werden viele Beispiele präsentiert, die die Verwendung des Tools demonstrieren. Diese Beispiele werden in einer Galerie nebeneinander angezeigt und mit Thumbnails versehen. Leider werden jedoch keine weiteren Informationen zu den einzelnen Beispielen angezeigt, sodass es schwierig ist, genau zu ermitteln, worum es bei einem bestimmten Beispiel geht. Die drei folgenden herausgesuchten Beispiele aus dieser Liste sollen kurz die verschiedenen Möglichkeiten des Tools veranschaulichen: (three.js, 2023)

ifc.js ist eine Open-Source-Bibliothek für die Darstellung von IFC-Modellen im Webbrowser. IFC ist ein offener internationaler Standard für den Austausch von Bau- und Gebäudedaten zwischen verschiedenen CAD-Programmen und anderen Anwendungen im Bausektor. Mit ifc.js können IFC-Modelle in gängigen Webbrowsern wie Chrome, Firefox oder Safari dargestellt und interaktiv bearbeitet werden. Die Verwendung von ifc.js ermöglicht es Architekten, Bauingenieuren, Planern und anderen Fachleuten im Bausektor, IFC-Modelle im Webbrowser darzustellen und zu analysieren, ohne spezielle CAD-Software installieren oder kaufen zu müssen. (IFC.js, 2022)

Ein weiteres Beispiel kommt aus dem Entertainment-Sektor. The Wilderness Downtown ist ein interaktives Musikvideo der kanadischen Indie-Rock-Band *Arcade Fire*. Es wurde in Zusammenarbeit mit Google Chrome entwickelt und nutzt Three.js, um visuelle Effekte zu erzeugen. Das Musikvideo ermöglicht es den Zuschauern, ihren eigenen Heimatort auf einer interaktiven Karte auszuwählen und sich dann von einer Vogelperspektive aus durch ihre Straßen zu bewegen, während das Lied im Hintergrund spielt. Dabei werden dynamische Effekte wie fallende Blätter und explodierende Gebäude erzeugt. (Gilbertson, 2010) (Milk, 2010)

Als drittes Beispiel für Projekte, die mithilfe von Three.js umgesetzt wurden, ist "Kaspersky Earth 2050" zu erwähnen. Diese interaktive Website ist ein interaktives Projekt, das von dem russischen Cybersicherheitsunternehmen Kaspersky ins Leben gerufen wurde. Das Projekt wurde entwickelt, um einen Blick in die Zukunft der Menschheit und der Welt im Jahr 2050 zu werfen und auf mögliche Bedrohungen und Herausforderungen hinzuweisen, die in den kommenden Jahrzehnten auftreten könnten. (Kaspersky Lab AO, 2023)

Kaspersky Earth 2050 besteht aus einer interaktiven Webseite und einer mobilen App. Auf der Webseite und in der App können Benutzer eine interaktive Karte der Welt im Jahr 2050 erkunden und verschiedene Szenarien und mögliche Entwicklungen auswählen, die sich auf den Klimawandel, die Technologieentwicklung, die Bevölkerungsentwicklung und andere Themen beziehen. Diese Prognosen könnten dazu beitragen, das Bewusstsein für die möglichen Auswirkungen von Umwelt- und Technologieentwicklungen auf die Menschheit und die Welt zu schärfen und die Diskussion darüber anzuregen, wie wir als Gesellschaft auf diese Herausforderungen reagieren können. (Simon-Lewis, 2017) (Kaspersky Lab AO, 2023)

# Implementierung

Nachdem die Einsatzmöglichkeiten, Technologien (Angular/Three.js) und der grobe Funktionsumfang der Software erfolgreich konzipiert wurden, liegt der Fokus nun auf der Implementierung und Kombination dieser verschiedenen Komponenten. In diesem Stadium wird das theoretische Konzept in die Praxis umgesetzt, indem die ausgewählten Technologien verwendet werden, um die gewünschten Funktionen zu entwickeln. Die Herausforderung besteht darin, die verschiedenen Bestandteile wie Angular-Komponenten und Three.js-Elemente nahtlos miteinander zu verbinden.

Bei der Implementierung des Projekts stehen insbesondere zwei Hauptbestandteile des Codes im Fokus. Angular dient dabei als grundlegender Rahmen, der die Struktur der Anwendung bereitstellt, während Three.js den Kern der 3D-Funktionalität bildet. Um diese beiden Komponenten zu integrieren, ist es zunächst erforderlich, ein neues Angular-Projekt einzurichten und anschließend Three.js darin zu implementieren. Für die Programmierung wird die Open-Source-Entwicklungsumgebung VS Code verwendet.

## Angular Implementierung

Um ein Angular-Projekt erfolgreich aufzusetzen, ist es erforderlich, die richtige Version von Node.js zu installieren. Für den Prototypen wurde die Long-Term-Support-Version 18.16.0 von Node.js verwendet. Diese Version wurde ausgewählt, da sie zum Zeitpunkt der Entwicklung die aktuellste stabile Version war und Langzeitunterstützung bietet.

Nachdem Node.js in der entsprechenden Version erfolgreich installiert wurde, konnte das Angular-Projekt gestartet werden. Hierbei wurde die neueste Version von Angular (Version 16.0.0) verwendet. Die Verwendung der neuesten Version ermöglicht es, von den neuesten Features und Verbesserungen zu profitieren und sicherzustellen, dass das Projekt auf dem aktuellen Stand der Technik ist. Um das Angular-Projekt in der gewünschten Version anzulegen, wurde der Node Package Manager (NPM) in der Version 9.6.6 verwendet.

Anschließend wurde Angular Material in der aktuellsten Version 16.1.5 dem erstellenten Projekt hinzugefügt. Für die Umsetzung des Prototypen ist hauptsächlich nur die *Grid-List* aus Angular-Materialnotwendig.

Angular Material Grid List ist eine Komponente, mit der Inhalte in einem flexiblen Rasterlayout angezeigt werden können. Die Komponente ermöglicht die Anzeige von Kacheln, Bildern oder anderen Inhalten in einem bereits mit Design versehenem Raster. Mit der Grid-List-Komponente können Inhalte einfach in Zeilen und Spalten organisiert werden. Es können die Anzahl der Spalten, die Höhe der Zeilen und der Abstand zwischen den Elementen angepasst werden. (Google LLC, 2023)

Diese ermöglicht die Unterteilung ähnlich einer Monitor-Wand. Um diese zu verwenden, muss das Modul in die *app-component.ts* Datei importiert werden. (Abbildung 1)

Nachdem dies erledigt ist, ist es möglich, HTML-Tags zu verwenden, um die Unterteilung und das Design der Tabelle zu steuern. Diese Tags werden in einer Schleife erzeugt und aktualisieren sich kontinuierlich, basierend auf der Anzahl der vorhandenen Kameras. Zusätzlich passt sich die Anordnung der Spalten automatisch der Anzahl der verfügbaren Kameras an. Zu Beginn nimmt die Spieleransicht die gesamte Anwendung ein, und wenn Kameras hinzugefügt werden, passt sich das Layout entsprechend an.

## Three.js Implementierung

In den genannten Angular-HTML-Tags werden die jeweiligen Ansichten der Kameras gerendert. Um dies zu ermöglichen, wird die Three.js-Bibliothek verwendet. Es gibt verschiedene Möglichkeiten, die Three.js-Bibliothek in ein Projekt einzubinden. Eine gängige Methode besteht darin, das Three.js-Paket als Node-Modul zu installieren. Dadurch kann die Bibliothek über den Paketmanager des Projekts, wie zum Beispiel npm (Node Package Manager), heruntergeladen und verwaltet werden. Durch den Import des Node-Moduls kann Three.js einfach in den Projektcode integriert werden, indem es als Abhängigkeit in den entsprechenden Dateien referenziert wird.

Eine alternative Methode besteht darin, Three.js über einen Web-Link zu importieren. Dabei wird die Three.js-Datei von einem externen Server geladen. Diese Datei kann entweder direkt von der Webseite abgerufen oder lokal im Projekt gespeichert werden. Wenn die Datei lokal gespeichert wird, kann sie in das Projektverzeichnis kopiert werden, um die Abhängigkeit von externen Servern zu reduzieren und die Verfügbarkeit der Datei sicherzustellen.

Sobald die Three.js-Datei im Projekt verfügbar ist, kann sie über den entsprechenden Pfad in den Projektdateien implementiert werden. Dies ermöglicht den Zugriff auf die Funktionalitäten von Three.js und die Nutzung der 3D-Grafikbibliothek für die Entwicklung der Anwendung.

Es gibt einen wichtigen Unterschied zwischen dem Node-Modul und der Datei. Das Node-Modul enthält Type Notationen, während die Objekte und Funktionen in der einzelnen Datei typenlos sind. Da Angular mit TypeScript und damit mit Type Notation arbeitet, kann die Verwendung der einzelnen Datei zu Problemen führen. Die meisten Beispiele, die von Three.js bereitgestellt werden, verwenden jedoch eine typenlose Programmierung, was die Migration zu TypeScript erschwert. In den verwendeten Beispielen werden Funktionen verwendet, die nicht erkannt werden, wenn mit Datentypen gearbeitet wird. In solchen Fällen muss der Code entsprechend angepasst werden.

Zusätzlich stimmt der Pfad, welcher für dem Import auf der Dokumentationsseite von Three.js zu finden ist, nicht mit dem Pfad überein, welcher in dem Angular-Projekt verwendet werden muss.

Nachdem Three.js implementiert wurde ging es darum, die Funktionalitäten umzusetzen. Der erste größere Schritt um dies zu erreichen, war einige relevante Beispiele auf der Three.js Seite miteinander zu kombinieren. Dies hat bereits ein User gemacht und ein Beispiel zur Verfügung gestellt (jdrew1303, 2020), welches die Grundlage des Prototypen bildet. Dieses Beispiel stellt die Basis für die Kameraansichten, deren Steuerung sowie die Unterteilung der Seite und das UI.

Bei diesem Beispiel und allen anderen Beispielen auf der Three.js-Website gab es jedoch zwei Herausforderungen, die gemeistert werden mussten, um effizient mit Angular arbeiten zu können. Angular basiert auf TypeScript, weshalb der JavaScript-Code aus den Beispielen in TypeScript umgewandelt werden musste, um eine optimale Programmiererfahrung zu gewährleisten. In den meisten Fällen wurde dieser Schritt problemlos von VS Code automatisch erledigt. Es gab jedoch einige Situationen, in denen kopierte JavaScript-Funktionen nicht reibungslos mit der importierten Three.js-Bibliothek, die auf TypeScript basierte, funktionierten. Daher musste der Code einiger Funktionen angepasst werden, um sicherzustellen, dass sie ordnungsgemäß in TypeScript funktionieren. Nachdem das Beispiel erfolgreich in Angular mit TypeScript zum Laufen gebracht wurde, folgte der nächste Schritt: das Laden eines 3D-Objekts in die Szene. Hierfür wurde ein Beispiel von Three.js verwendet, das es ermöglichte, ein glTF-Objekt in die Mitte der Szene zu laden und die Kamera um das Objekt herum zu bewegen. Auch dieses Beispiel musste an die Anforderungen von TypeScript angepasst werden, um eine nahtlose Integration in das Projekt zu gewährleisten.

Das 3D Objekt, welches für den Prototypen verwendet wird, ist ebenfalls eine kostenlos zur Verfügung gestellte *.blend* Datei (gerhald3d, 2019), welche anschließend über Blender in das glTF Format exportiert wurde.

Nachdem diese zwei grundlegenden Mechaninken eingebaut waren,

## Kamera Problem

Die verwendeten Kameras bieten nach eigenen Versuchen zwei Möglichkeiten, die Blickrichtung der Kamera zu definieren. Eine Option besteht darin, ein "Target" festzulegen, um das sich die Kamera bei Bewegung bewegt. Das Target wird jedoch nicht direkt in der Kamera selbst gesetzt, sondern in einem *OrbitControls*-Objekt, das die Steuerung der Kamera über Mausbewegungen ermöglicht.

Es besteht jedoch der Wunsch, dass die Kamera nicht um ein Objekt herumdreht, sondern um ihren eigenen Ursprung. Daher wurde das Target des *OrbitControls*-Objekts entsprechend des Kamera-Ursprungs gesetzt. Eine umfangreiche Recherche hat leider keine alternative Lösung ergeben.

Dies führt jedoch dazu, dass die Kameras nicht mehr gedreht werden können bzw. die Steuerung des "Controls"-Objekts nicht mehr funktioniert. Eine Möglichkeit, dies zu umgehen, besteht darin, eine der Target-Koordinaten um einen sehr kleinen Wert zu ändern, sodass der Target-Ursprung nicht mehr exakt dem Kamera-Ursprung entspricht. Allerdings führt dies dazu, dass die Kamera um 90 Grad auf der Y-Achse gedreht wird und diese Rotation nicht beim ersten Rendern geändert werden kann. Der Wert ist immer ein negatives π/2. Folglich wird immer eine falsche Perspektive angezeigt, die um 90 Grad verdreht ist.

Zusätzlich gibt es die Funktion *lookAt()*, die lediglich die Blickrichtung der Kamera festlegt, ohne den Mittelpunkt der Kamerabewegung zu ändern. Um das vorherige Problem zu minimieren, wird im Prototypen beim Platzieren der Kameras der Spieler als Blickpunkt festgelegt. Bei jedem Rendervorgang wird die Kamera entsprechend auf den Spieler ausgerichtet, wobei die Kamerasteuerung deaktiviert ist. Über die GUI kann jedoch die Kamera vom Spieler getrennt werden, was dann wieder eine Steuerung ermöglicht. Beim Verwenden der Kamerasteuerung und der Änderung mindestens einer der Rotationskoordinaten springt die Kamera jedoch beim ersten Mal auf den vorher erläuterten falschen Wert für die Y-Achse (90 Grad). Dieses Phänomen tritt jedoch nur einmal auf. Nachdem es einmal durchlaufen wurde, tritt das Problem nicht mehr auf und die Kameras können beliebig gedreht und an den Spieler geheftet werden, ohne dass die Kamera diesen falschen Y-Wert noch einmal verwendet.

Ein weiteres Problem mit den Kameras besteht darin, die Kameras nach einer kurzen Zeit die Blickrichtung ändern. Bei der Platzierung einer neuen Kamera zeigt diese recht genau denselben Bildausschnitt wie die der Spieler-Perspektive. Nach einem kurzen Zeitraum von ungefähr einer Sekunde wechselt die Kameraperspektive allerdings immer auf das Zentrum des Welt-Koordinatensystems.

# Komponenten Entwurf

# Evaluierung/Diskussion

War es eine kluge Entscheidung, eine Javascript-basierte 3D-Bibliothek mit einem auf Typescript basierenden Web-Framework zu verknüpfen, um eine Software für 3D-Kameraansichten zu erstellen? Durch die bereits vorhandenen Kenntnisse in Angular zu Beginn des Projektes entfiel die Notwendigkeit, sich in ein anderes Framework einzuarbeiten. Außerdem erwies sich Visual Studio Code als nützliches Werkzeug beim Übersetzen von Javascript zu Typescript, was einen reibungslosen Start ermöglichte.

Die Evaluation der Verwendung von React.js als Fundament im Vergleich zu Angular ist durchaus angebracht, da die Basis in JavaScript (Meta Open Source, 2023), wie bereits im Namen erwähnt, auf eine nahtlose Integration ohne Notwendigkeit einer Übersetzung verweist. Angular bietet zwar auch ein Framework auf JavaScript-Basis an, allerdings ist hier der Support im Januar 2022 abgelaufen (Thompson, 2022).

# Zusammenfassung

Videoüberwachungssysteme gewinnen zunehmend an Bedeutung für Unternehmen und öffentliche Einrichtungen. Der weltweite Markt für solche Systeme verzeichnet ein deutliches Wachstum. Bevor Unternehmen in Videoüberwachung investieren, müssen sie sicherstellen, dass sie die geltenden gesetzlichen Bestimmungen einhalten, die den Einsatz von Kameras regeln. Allerdings reicht allein die Videoüberwachung nicht mehr aus, um Gewaltkriminalität effektiv zu bekämpfen. Es bedarf zusätzlicher Maßnahmen wie einer besseren Beleuchtung und dem Einsatz von Sicherheitspersonal, um die Wirksamkeit der Überwachung zu erhöhen.

Effizienz und Kostenersparnis sind weitere Gründe, die für den Einsatz von Videoüberwachungssystemen in Unternehmen sprechen. Trotz Bedenken hinsichtlich der Privatsphäre zeigt die Forschung, dass die Mehrheit der Bevölkerung Videoüberwachung in öffentlichen Bereichen akzeptiert.

Angesichts dessen entstand dieses Projekt, da die Software der kooperierenden Firma dieser Thesis keine Möglichkeit bietet, Echtzeitänderungen in 3D darzustellen. Das Ziel war es, ein Projekt zu entwickeln, das sich in ein bestehendes System integrieren lässt und die interaktive 3D-Darstellung von Objekten ermöglicht. Zu diesem Zweck wurden drei Use-Cases entwickelt, um die Softwareplanung zu verbessern: die Schulung des Wachpersonals in einem Einkaufszentrum, die Verfolgung einer fahrenden Kamera und die Echtzeit-Verfolgung eines bestimmten Objekts. Die entwickelte Software bietet dem Benutzer die Möglichkeit, das 3D-Objekt aus verschiedenen Blickwinkeln zu betrachten und sich frei durch die virtuelle Umgebung zu bewegen.

Die Verwendung von Webtechnologien ermöglicht eine nahtlose Integration in bestehende Anwendungen von Funkwerk. Die Benutzeroberfläche wird in einem Gitterlayout angeordnet, um die gleichzeitige Darstellung einer größeren Anzahl von Kamerastreams zu ermöglichen. Die Software ermöglicht es dem Benutzer, die Einstellungen für jede Kamera einfach anzupassen und erstellte Kameras bei Bedarf zu löschen.

Für die prototypische Umsetzung werden vereinfachte Kameras ohne optische Fehler oder Begrenzungen verwendet. Der einzige relevante Parameter ist die Brennweite, mit der der Bildausschnitt gewählt werden kann. Die 3D-Objekte im Projekt werden im glTF-Dateiformat bereitgestellt. glTF bietet eine effiziente Möglichkeit, 3D-Inhalte zu übertragen und zu laden und ermöglicht die Darstellung vollständiger 3D-Szenen in Webanwendungen.

Um das Projekt umsetzten zu können, müssen die zwei Technologien Angular und Three.js nahtlos miteinander verbunden werden. Angular bildet den grundlegenden Rahmen für die Anwendungsstruktur, während Three.js die 3D-Funktionalität bereitstellt.

Die Implementierung erfolgt mit der Installation von Node.js, um das Angular-Projekt zu erstellen, und der Nutzung von Angular Material für die Darstellung der Kameravorschau. Three.js wird in das Angular-Projekt integriert, indem entweder eine Bibliothek heruntergeladen oder ein Web-Link genutzt wird. Der JavaScript-Code aus den Three.js-Beispielen wird in TypeScript umgewandelt, um die Programmiererfahrung zu verbessern. Das 3D-Objekt für den Prototyp wird im glTF-Format bereitgestellt, nachdem es aus einer .blend-Datei exportiert wurde.

Bei der Umsetzung der Kameras treten zwei Herausforderungen auf: die Definition der Blickrichtung durch ein "Target" oder die Drehung um den eigenen Ursprung. Die gewünschte Anforderung, das Drehen um die eigene Achse, erfordert Anpassungen, um die gewünschte Steuerung zu erreichen. Ein weiteres Problem tritt auf, wenn die Kameras nach kurzer Zeit ihre Blickrichtung ändern und auf das Zentrum des Welt-Koordinatensystems wechseln. Dieses Verhalten muss untersucht und behoben werden, um die gewünschte Funktionalität der Kameras zu gewährleisten.

# Ausblick/ Weiterentwicklungsmöglichkeiten

Neben der grundlegenden Funktionalität, dreidimensionale Objekte nahtlos in eine Web-Oberfläche zu integrieren und eine interaktive Navigation durch sie zu ermöglichen, bietet die Software auch die Möglichkeit, weitere Funktionen hinzuzufügen. Diese Funktionen können jedoch eine hohe Komplexität aufweisen, die in dieser Arbeit nur oberflächlich behandelt wird, um einen groben Überblick zu vermitteln. Es ist wichtig anzumerken, dass diese zusätzlichen Funktionen eine breite Palette von Möglichkeiten abdecken können, je nach den Anforderungen und Zielen des Projekts. Die genaue Auswahl und Umsetzung dieser Funktionen hängt von den spezifischen Anforderungen des Projekts ab und erfordert möglicherweise weitere Entwicklungsarbeit und Integration mit anderen Technologien und APIs. Die Entscheidung über die Integration zusätzlicher Funktionen sollte sorgfältig abgewogen werden, um den Umfang des Projekts, die verfügbaren Ressourcen und die Projektziele in Einklang zu bringen. Nachfolgend werden drei Beispiele von Erweiterungsmöglichkeiten näher betrachtet.

## Verdeckung

Diese Funktion besteht darin, das gesamte Objekt rot anzuzeigen oder rot leuchten zu lassen, um es visuell hervorzuheben. Gleichzeitig werden die erfassten Bereiche der Kameras und anderer Kameras in grün eingefärbt.

Diese Erweiterung ermöglicht es dem Benutzer sofort zu erkennen, welche Bereiche des 3D-Objekts von den Kameras überwacht werden und welche nicht. Durch die rote Farbgebung oder das Hervorheben des gesamten Objekts wird eine visuelle Aufmerksamkeit erzeugt. Gleichzeitig wird die Benutzerfreundlichkeit verbessert, da der Benutzer auf einen Blick sehen kann, welche Bereiche aktuell erfasst werden.

Sobald der Benutzer das 3D-Objekt erkundet und sich bewegt, sind die erfassten Bereiche der Kameras in grün eingefärbt. Dies ermöglicht es dem Benutzer, schnell zu erkennen, welche Bereiche von den Kameras überwacht werden und bekommt somit in Echtzeit visuelle Rückmeldungen über die aktuelle Überwachungsabdeckung.

Durch diese Funktion wird die Benutzererfahrung verbessert, da der Benutzer auf einen Blick erkennen kann, welche Bereiche innerhalb des 3D-Objekts von den Kameras erfasst werden und somit in den entsprechenden Perspektiven angezeigt werden. Dies trägt zur Orientierung und effektiven Nutzung der Software bei.

Die Umsetzung dieser Funktion erfordert die Anpassung der visuellen Darstellung des 3D-Objekts sowie die Integration von Logik zur Echtzeitüberwachung und Einfärbung der erfassten Bereiche. Hierbei können Techniken wie Raytracing verwendet werden, um die erfassten Bereiche der Kameras zu bestimmen und sie grafisch hervorzuheben. Dabei senden die Kamera Strahlen aus, welche sich wie Licht bewegen und sobald diese auf etwas treffen, wird das rot eingefärbt.

Insgesamt eröffnet diese Erweiterung der Software neue Möglichkeiten für den Benutzer, das 3D-Objekt intuitiver zu erkunden und die Überwachungsbereiche der Kameras besser zu verstehen. Durch die visuelle Hervorhebung des gesamten Objekts in Rot und die grüne Einfärbung der erfassten Bereiche wird die Benutzerfreundlichkeit und Effektivität der Software verbessert.

## Mathematische Algorithmen

Eine große mögliche Erweiterung der Software besteht darin, mathematische Algorithmen zu implementieren, die die optimale Position und Ausrichtung der Kameras bestimmen. Dies kann basierend auf den spezifischen Anforderungen des geladenen 3D-Objekts und der zu erfassenden Fläche erfolgen.

Ein wichtiger Aspekt bei der Umsetzung dieser Funktionalität ist die Berücksichtigung einer maximal verfügbaren Anzahl an Kameras für das System oder des Kunden. Die Software könnte daher mathematische Berechnungen und Optimierungsalgorithmen verwenden, um die minimale oder maximale Anzahl an Kameras zu ermitteln, die erforderlich ist, um eine umfassende Abdeckung des 3D-Objekts zu gewährleisten. Hierbei werden Faktoren wie die Größe des Objekts und die gewünschte Detailgenauigkeit berücksichtigt.

Ein weiterer wichtiger Faktor ist die Bestimmung optimaler Brennweiten für die Kameras. Mathematische Modelle und Analyseverfahren können verwendet werden, um die optimalen Brennweiten zu ermitteln, die das Sichtfeld der Kameras effizient abdecken. Dabei können verschiedene Parameter einbezogen werden, wie beispielsweise die Größe des zu erfassenden Objekts, die Abstände zwischen den Kameras und die gewünschte Bildqualität. Durch die Anpassung der Brennweiten können die Kameras optimal auf das 3D-Objekt ausgerichtet werden, um eine bestmögliche Sichtbarkeit und Detailgenauigkeit zu gewährleisten.

Allerddings kann auch das Gegenteil der Fall sein und es muss mit vorhandenen und nicht änderbaren Brennweiten gearbeitet werden. Hierbei muss der Algorithmus mit den fest vorgegeben Werten arbeiten und kann diese nicht für eine optimale Platzierung anpassen.

Die Herausforderung bei der Implementierung dieser mathematischen Algorithmen besteht darin, die verschiedenen Variablen und Parameter in die Berechnungen einzubeziehen, um eine optimale Lösung zu finden. Die Software müsste eine umfassende Analyse durchführen und Optimierungstechniken wie beispielsweise genetische Algorithmen oder Optimierung auf Basis künstlicher Intelligenz einsetzen. Dadurch könnte eine effiziente und dennoch vollständige Kameraabdeckung erreicht werden.

Durch die Umsetzung dieser Erweiterung eröffnen sich zahlreiche Möglichkeiten. Zum einen ermöglicht die optimale Positionierung und Ausrichtung der Kameras eine verbesserte Überwachung des 3D-Objekts aus verschiedenen Blickwinkeln und somit auch eine bessere Planbarkeit. Zum anderen können durch die effiziente Nutzung der Kameras Ressourcen eingespart werden, was insbesondere bei Systemen mit begrenzten Hardwarekapazitäten von Vorteil ist.

Darüber hinaus ermöglicht die Implementierung dieser mathematischen Algorithmen eine automatisierte Anpassung der Kameras an verschiedene Szenarien und Objekte. Dies spart dem Benutzer Zeit und Aufwand bei der manuellen Platzierung und Ausrichtung der Kameras. Die Software kann die optimale Konfiguration der Kameras in Echtzeit berechnen, basierend auf den aktuellen Anforderungen des 3D-Objekts oder der zu erfassenden Fläche.

## Weitere Person kann sich durch das 3D Objekt bewegen

Eine weiteres großes Feature Update könnte die Implementierung einer weiteren virtuellen Person sein, welche programmiert werden kann um bestimmte Routen automatisch abzulaufen.

Für diese Implementierung, sind mehrere Schritte erforderlich. Zunächst muss ein 3D-Modell für die zweite Person erstellen oder auf vorhandene Modelle zurückgegriffen werden. Dieses Modell wird in einem geeigneten 3D-Format erstellt und in die Software integriert, um es korrekt in der Szene darzustellen. Am besten wäre auch hier das glTF Format um eine Einheitlichkeit zu gewährleisten.

Sobald das 3D-Modell integriert wurde, kann mit der Programmierung der Bewegungsrouten begonnen werden. Es gibt hierbei die Möglichkeit, vordefinierte Pfade zu verwenden oder Wegpunkte festzulegen, die die Person ablaufen soll. Durch die Implementierung einer entsprechenden Funktion können diese Bewegungsrouten definieren werden oder Punkte gesetzt werden.

Um dem Benutzer die Steuerung der Bewegungsrouten zu ermöglichen, kann eine Benutzerschnittstelle implementiert werden. Diese Schnittstelle ermöglicht es dem Benutzer, Wegpunkte festzulegen, die Geschwindigkeit oder Richtung der Bewegung anzugeben und die Bewegung der zweiten Person zu starten oder zu stoppen. Weiterhin soll auch die Blickrichtung der Person angepasst werden um genau sehen zu können, was die virtuelle Person auf ihrer Route erkennen kann.

Damit die zweite Person mit der virtuellen Umgebung interagieren kann, ist die Implementierung einer Kollisionserkennung erforderlich. Dadurch kann die Person beispielsweise Hindernissen ausweichen oder auf bestimmte Ereignisse reagieren. Das wäre vor allem wichtig, wenn die Route nur über die Wegpunkte geregelt wird. Hier muss der Charakter selbstständig den Weg zum nächsten Punkt finden und eventuellen Hindernissen ausweichen. Bei der kompletten Pfad-Vorgabe ist das dynamische Anpassen der Route zwar nicht nötig, allerdings wird auch hier die Kollision benötigt um sich an den Untergrund des 3D Objektes anpassen zu können.

Die Implementierung dieser Funktion eröffnet eine Vielzahl von Möglichkeiten. Zum Beispiel können damit virtuelle Rundgänge erstellt werden, bei denen eine zweite Person den Benutzer durch die 3D-Umgebung führt. Dies kann besonders nützlich sein, um komplexe Szenarien oder Produktpräsentationen zu visualisieren. Weiterhin kann damit auch anschaulich und automatisiert dargestellt werden, wie eine Person bestimmte Wege war nimmt und ob es

# Literaturverzeichnis

Axis Communications AB. (2021). *Sicherheitsgefühl in deutschen Städten: Studie zeigt hohe Akzeptanz von Smart City-Technologien.* Ismaning.

Blender Foundation. (18. 07 2023). *The Freedom to Create*. Von https://www.blender.org/: https://www.blender.org/about/ abgerufen

DDI Development. (01 2019). *The Pros and Cons of choosing Angular for web app development*. Abgerufen am 13. 04 2023 von https://ddi-dev.com/blog/programming/pros-and-cons-of-angular-web-app-development/

*examples*. (2023). Abgerufen am 02. 04 2023 von threejs.org: https://threejs.org/examples/

*Fundamentals*. (kein Datum). Abgerufen am 16. 04 2023 von three.js: https://threejs.org/manual/#en/fundamentals

gerhald3d. (29. 11 2019). GameReady Cottage 3D-Modell. free3d.com. Abgerufen am 12. 04 2023 von https://free3d.com/3d-model/gameready-cottage-free-163528.html

Gilbertson, S. (17. 07 2010). Behind the Scenes: Coding 'The Wilderness Downtown'. *WIRED*.

GitHub, Inc. (31. 03 2023). *three.js*. Abgerufen am 02. 04 2023 von Github: https://github.com/mrdoob/three.js

Google LLC. (2023). *Components*. Abgerufen am 15. 07 2023 von Angular Material: https://material.angular.io/components/categories

Google LLC. (2023). *Grid List*. Abgerufen am 20. 07 2023 von Angular Material: https://material.angular.io/components/grid-list/overview

Gregory, J. (2018). *Game Engine Architecture* (Bd. 3). Boca Raton: CRC Press.

Harris, R. (26. 11 2016). *Frameworks without the framework: why didn't we think of this sooner?* Abgerufen am 21. 03 2023 von svelte.dev: https://svelte.dev/blog/frameworks-without-the-framework

Herzog, M. (23. 03 2021). *Owners*. Abgerufen am 02. 04 2023 von three.js: https://github.com/mrdoob/three.js/wiki/Owners

IDS Imaging Development Systems GmbH. (2009). *High Dynamic Range Imaging: Images and Sensors Fundamentals, Method of Functioning, Application.* Obersulm.

IFC.js. (2022). *Introduction*. Abgerufen am 14. 04 2023 von IFC.js: https://ifcjs.github.io/info/docs/introduction

International Organization for Standardization & International Electrotechnical Commission. (07 2022). *Information technology — Runtime 3D asset delivery format — Khronos glTF™ 2.0* (1 Ausg.). Abgerufen am 06. 05 2023 von https://www.iso.org/standard/83990.html

intersoft consulting services AG. (kein Datum). *Was ist die Datenschutz-Grundverordnung?* Abgerufen am 13. 04 2023 von Datenschutz, IT‑Sicherheit und IT‑Forensik: Wir halten Sie auf Kurs.: https://www.intersoft-consulting.de/infos/datenschutz-grundverordnung-dsgvo/

IPICA LLC. (01 2023). *Videoüberwachung Planung und Projektierung Tool*. Abgerufen am 15. 03 2023 von Video Surveillance Design Apps | JVSG: https://www.jvsg.com/

jdrew1303. (13. 09 2020). *Three.js - Multiple Cameras*. Abgerufen am 05. 05 2023 von codepen.io: https://codepen.io/jdrew1303/pen/poyVOyG

Kaspersky Lab AO. (2023). *About*. Abgerufen am 15. 04 2023 von Earth 2050: https://2050.earth/about

Kaspersky Lab AO. (2023). *PREDICTIONS*. Abgerufen am 15. 04 203 von Earth 2050: https://2050.earth/predictions

Markt für Videoüberwachungssysteme - Wachstum, Trends, Auswirkungen von Covid-19 und Prognosen (2022 – 2027). (2022). Abgerufen am 22. 05 2023 von Mordor Intelligence: https://www.mordorintelligence.com/de/industry-reports/global-video-surveillance-market-industry

Meta Open Source. (21. 04 2023). *React*. Abgerufen am 24. 07 2023 von Github: https://github.com/facebook/react

Milk, C. (Regisseur). (2010). *The Wilderness Downtown* [Kinofilm]. Von http://www.thewildernessdowntown.com/. abgerufen

Piza, E. L., Welsh, B. C., Farrington, D. P., & Thomas, A. L. (24. 03 2019). *CCTV surveillance for crime prevention: A 40-year systematic review with meta-analysis.* Criminology & Public. Von https://onlinelibrary.wiley.com/doi/10.1111/1745-9133.12419 abgerufen

*Responsive Design*. (kein Datum). Abgerufen am 16. 04 2023 von three.js: https://threejs.org/manual/#en/responsive

Riedl, D. (2018). Akzeptanz von Videoüberwachung steigt. *Security Insider*. Abgerufen am 22. 05 2023 von https://www.security-insider.de/akzeptanz-von-videoueberwachung-steigt-a-777706/

Russel, M. (2019). Supporting Decision Makers with Use Cases; Case Study Results. *Procedia Computer Science 153*, S. 294-300.

Simon-Lewis, A. (05. 04 2017). Futuristic map shows what Earth could look like in 2050. *WIRED*.

Stack Overflow. (01. 06 2022). *2022 Developer Survey*. Abgerufen am 21. 03 2023 von survey.stackoverflow.com: https://survey.stackoverflow.co/2022

Thompson, M. (11. 01 2022). *Discontinued Long Term Support for AngularJS*. Abgerufen am 24. 07 2023 von Medium: https://blog.angular.io/discontinued-long-term-support-for-angularjs-cc066b82e65a

*three.js*. (2023). Abgerufen am 16. 04 2023 von three.js: https://threejs.org/

*What is Angular?* (28. 02 2022). Abgerufen am 18. 03 2023 von Angular: https://angular.io/guide/what-is-angular

Оглукян, A. K. (24. 04 2020). OVERVIEW OF THE ANGULAR FRAMEWORK: PROS AND CONS. *ТЕОРИЯ И ПРАКТИКА МОДЕРНИЗАЦИИ НАУЧНОЙ ДЕЯТЕЛЬНОСТИ В УСЛОВИЯХ ЦИФРОВИЗАЦИИ*, S. 33-37.

# Bilder



Abbildung Import Material-Componente in app-component.ts

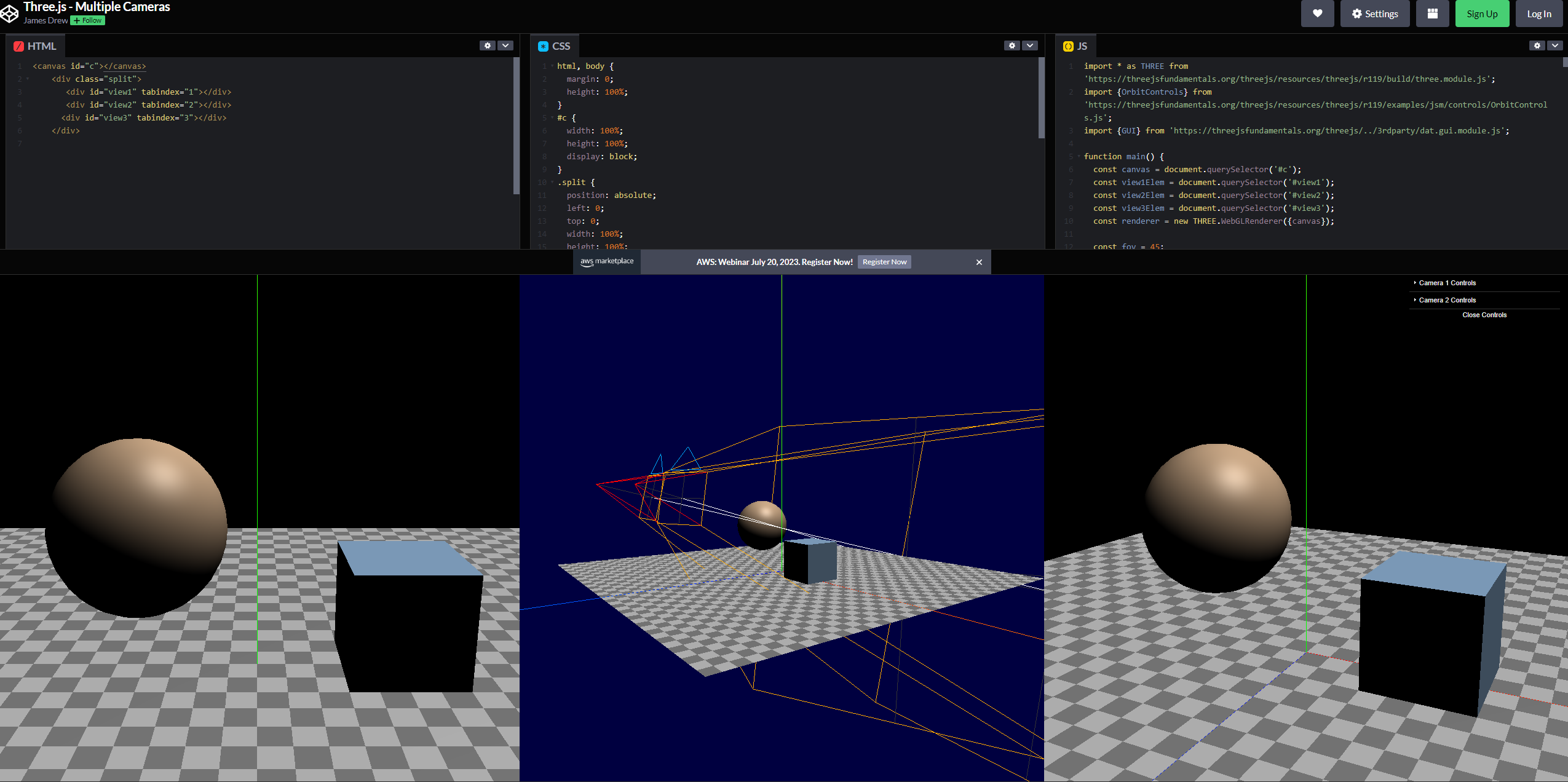


Abbildung 2 Grundlage des Prototypen

# Stichpunkte

* Einfaches UXD
* Node.js 18.16.0 LTS
* Solid Prinzipien umsetzten (klassen ausgliedern)
* Npm 9.6.6
* THREE node Module hat nicht alle Inhalte ( doch hat es, aber die script Datei hat nahezu alles in „any“ und damit werden funktionen verwendet, welche mit der type-notation nicht funktionieren)
* Angular 16.0.0
* Vorlage Code (jdrew1303, 2020)
* Import Loader falsch auf Website dokumentiert bei Verwendung von Node modules

### Allgemein

Funktionen:

* Installation von Kameras in strategischen Bereichen wie Eingängen, Korridoren, Treppenhäusern und Aufzügen
* Live-Übertragung von Video-Feeds auf einen zentralen Server, der von Sicherheitspersonal überwacht wird
* Aufzeichnung von Video-Feeds für spätere Überprüfung und Nachverfolgung von Vorfällen
* Integration mit einem Alarm-System, das ausgelöst wird, wenn verdächtige Aktivitäten erkannt werden
* Benachrichtigung von Sicherheitspersonal und/oder der Polizei im Falle eines Vorfalls
* Zugriff auf Live-Feeds und aufgezeichnete Video-Feeds durch autorisierte Personen wie das Sicherheitspersonal und das Management

### Kindergarten

Funktionen:

* Installation von Kameras in verschiedenen Bereichen der Einrichtung, wie z.B. Spielbereichen, Schlafbereichen, Küchenbereichen und Eingängen
* Live-Übertragung von Video-Feeds auf einen zentralen Server, der von qualifiziertem Personal überwacht wird
* Möglichkeit für Eltern, sich über eine sichere Verbindung mit den Live-Feeds der Kameras zu verbinden und die Aktivitäten ihrer Kinder zu überwachen
* Aufzeichnung von Video-Feeds für spätere Überprüfung und Nachverfolgung von Vorfällen
* Integration mit einem Alarm-System, das ausgelöst wird, wenn verdächtige Aktivitäten oder Gefahren für Kinder erkannt werden
* Benachrichtigung des Personals und der Eltern im Falle eines Vorfalls oder einer Gefahrensituation
* Zugriff auf Live-Feeds und aufgezeichnete Video-Feeds nur durch autorisierte Personen wie das qualifizierte Personal und die Eltern

### KFZ-Halle

Die Installation einer Kamera-Überwachungssoftware in einer KFZ Halle bringt zahlreiche Vorteile für den Betrieb. Kameras können in verschiedenen Bereichen wie Eingängen, Ausfahrten, Parkplätzen und Arbeitsbereichen installiert werden, um eine umfassende Überwachung des Standorts zu gewährleisten. Die Live-Übertragung der Video-Feeds auf einen zentralen Server ermöglicht qualifiziertem Personal, die Feeds in Echtzeit zu überwachen und aufgezeichnete Video-Feeds später zu überprüfen.

Das Personal kann auf einem Monitor die Live-Feeds der Kameras überwachen und bei Bedarf in Echtzeit reagieren. Im Falle von Vorfällen wie Unfällen oder Diebstählen kann das aufgezeichnete Videomaterial später überprüft werden, um den Vorfall zu analysieren und mögliche Täter zu identifizieren. Die Überwachungssoftware kann auch mit einem Alarm-System integriert werden, das ausgelöst wird, wenn verdächtige Aktivitäten oder Gefahren für Fahrzeuge, Personal oder Besucher erkannt werden. Dadurch kann das Personal schnell reagieren und potenzielle Bedrohungen frühzeitig erkennen.

Im Falle eines Vorfalls oder einer Gefahrensituation wird das Personal automatisch benachrichtigt, um schnell reagieren zu können. Die Zugriffsrechte auf Live-Feeds und aufgezeichnete Video-Feeds sind streng auf autorisiertes Personal beschränkt, um die Privatsphäre der Mitarbeiter und Kunden zu schützen und den Missbrauch der Kameras zu verhindern.

Insgesamt könnte die Installation einer Kamera-Überwachungssoftware in einer KFZ Halle ein hohes Maß an Sicherheit und Schutz für Mitarbeiter, Kunden und das Unternehmen selbst bieten. Die Überwachungssoftware ermöglicht es, potenzielle Vorfälle zu erkennen und schnell zu reagieren, was letztendlich dazu beiträgt, den Betrieb reibungsloser und sicherer zu gestalten.

### Krankenhausüberwachung

Die Installation einer Kamera-Überwachungssoftware in einem Krankenhaus bietet zahlreiche Vorteile für die Sicherheit und den Schutz von Patienten, Personal und Besuchern. Kameras können in verschiedenen Bereichen des Krankenhauses installiert werden, wie Eingängen, Fluren, Aufzügen, Notaufnahmen und Intensivstationen, um eine umfassende Überwachung des Standorts zu gewährleisten. Die Live-Übertragung der Video-Feeds auf einen zentralen Server ermöglicht qualifiziertem Personal, die Feeds in Echtzeit zu überwachen und aufgezeichnete Video-Feeds später zu überprüfen.

Das Personal kann auf einem Monitor die Live-Feeds der Kameras überwachen und bei Bedarf in Echtzeit reagieren. Im Falle von Vorfällen wie Diebstählen oder Unfällen kann das aufgezeichnete Videomaterial später überprüft werden, um den Vorfall zu analysieren und mögliche Täter zu identifizieren. Die Überwachungssoftware kann auch mit einem Alarm-System integriert werden, das ausgelöst wird, wenn verdächtige Aktivitäten oder Gefahren für Patienten, Personal oder Besucher erkannt werden. Dadurch kann das Personal schnell reagieren und potenzielle Bedrohungen frühzeitig erkennen.

Im Falle eines Vorfalls oder einer Gefahrensituation wird das Personal automatisch benachrichtigt, um schnell reagieren zu können. Die Zugriffsrechte auf Live-Feeds und aufgezeichnete Video-Feeds sind streng auf autorisiertes Personal beschränkt, um die Privatsphäre der Patienten und deren Schutz zu gewährleisten. Es kann auch der Zugriff auf bestimmte Bereiche wie z.B. Patientenzimmer eingeschränkt werden, um die Privatsphäre der Patienten zu schützen.

Insgesamt bietet die Installation einer Kamera-Überwachungssoftware in einem Krankenhaus ein hohes Maß an Sicherheit und Schutz für Patienten, Personal und Besucher. Die Überwachungssoftware ermöglicht es, potenzielle Vorfälle zu erkennen und schnell zu reagieren, was letztendlich dazu beiträgt, den Betrieb reibungsloser und sicherer zu gestalten.

* Wachmann ergänzt Kameras
* Sichtbarkeit der Kameras kann von einer virtuellen Person überprüft werden
* Hängende Kameras können gesetzt werden ohne Stange modellieren zu müssen
* Nice to Haves vs essentiell
* Schulung für Wachmann => wenn dynamisches Objekt, dann da langlaufen
* Kameras nur bei Kollision platzieren (aber bei freier Setzung auch im Raum schwebend einsetzbar
* Roboter mit Kamera fahren lassen (Lüftungsschächte/für Menschen unzugängliche Bereiche)
* Kameras bewusst Sichtbar/Unsichtbar platzieren (Personen suchen gezielt nach Sicherheit/gezielt nach keiner Überwachung)
* Drohnenkamera zeigt Sicht, welche Menschen sich nur vorstellen können (erst nach Kamerainstallation sichtbar)
* Brennweite der Kameras schon von Vornherein feststellbar / auswählbar ohne Mathematische Kalkulationen
* Tatsächliche Sichtbarkeit von Bereichen überprüfen (Maschendrahtzaun gibt mathematisch perfekte Sicht, aber nicht tatsächliche Sicht)
* Genau da, wo ich gerade hin schaue eine Kamera platzieren
* Farben sehen (bewusste Täuschung?)
* Licht (Ausleuchtung/Sonne)
* Akustik (Gefahren/Lautsprecherdurchsagen)
* Nachbarn/Straße/Verbotene Bereiche, welche trotzdem überwacht werden (unabsichtlich)
* Was ist in 10 Jahren (Licht / Baum wächst/ neues Gebäude wird gebaut) (Winter/Sommer)
* Spiegelkasten?

Blickpunkte und Erfassungsbereich von Kameras spielen bei der Positionierung eine wichtige Rolle – in unterschiedlichen Domänen von Überwachungssystemen bis Computergrafik. Im Rahmen der Thesis wird ein System konzipiert und prototypisch implementiert, dass es Nutzern ermöglicht in einem dreidimensionalen Modell, beispielsweise von Gebäuden, Kameras zu positionieren und Sichtbereiche zu erfassen (Position, Winkel, Blickrichtung, etc.).