



ARRaceTrackRC

Ein innovatives und fesselndes AR-Erlebnis im Bereich der ferngesteuerten
Fahrzeuge

Wintersemester 2023/2024

Jakob Lingel - 83275

Nicolas Ostermann - 82796

Lea-Marie Kindermann – 82789

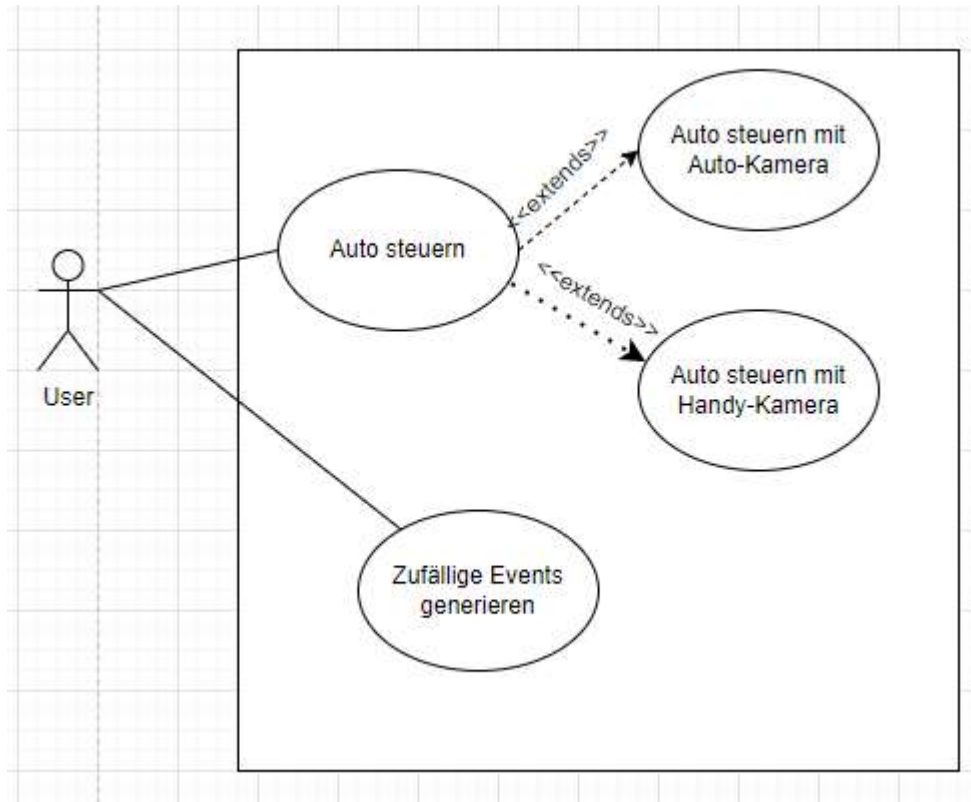
Jenny Jost - 83272

Inhalt

Project overview and goal.....	1
Use Case Diagram	1
Liste von Fähigkeiten und Features	2
Scenario and Functionality Description	2
Auto steuern	2
Videoübertragung über die Handykamera.....	3
Videoübertragung über die Kamera am Auto	3
Events	4
Architecture (internal)	5
Klassendiagramm.....	5
Lessons Learned	6
as a team	6
Individually	7
Appendix.....	11
19.10.2023CONOPS MS Report.....	11
16.11.2023 Tech Demo MS Report.....	11
30.11.2023 Architectural Spike MS Report	12
21.12.2023 Alpha MS Report	12
18.01.2024 Beta MS Report.....	13

Project overview and goal

Use Case Diagram



Der erste Use Case ermöglicht es dem Benutzer, das Fahrzeug direkt über die im Auto integrierte Kamera zu steuern.

Ein weiteres Use Case ist der AR-Modus, der die Handykamera des Benutzers nutzt. In diesem Modus verschmilzt die virtuelle Welt des Autos nahtlos mit der realen Umgebung des Benutzers.

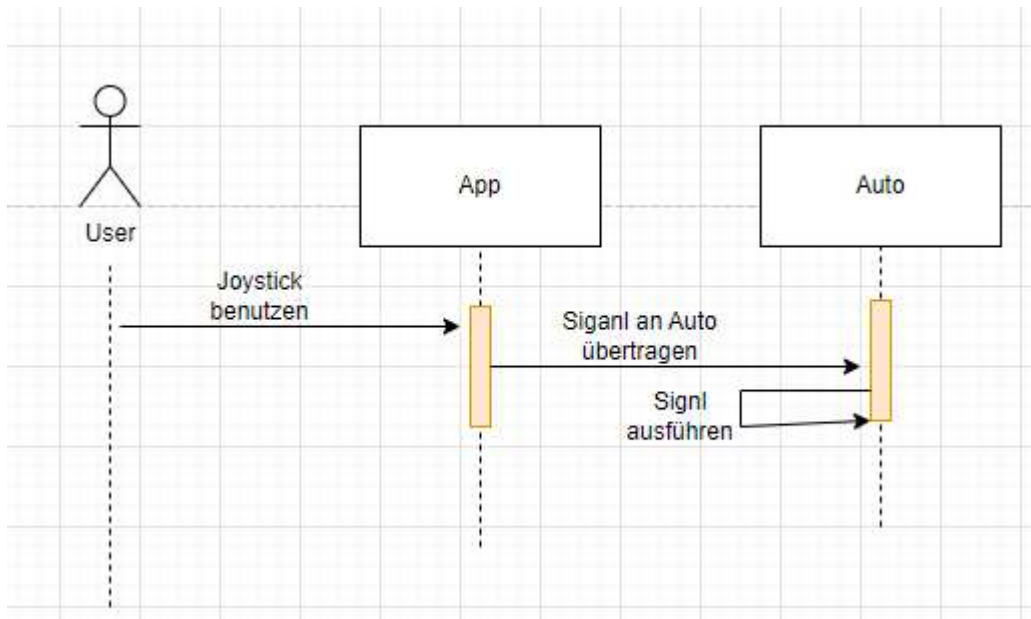
Ergänzend zu diesen Kernfunktionen sorgt das System für zusätzliche Spannung durch die Generierung zufälliger Events. Diese Ereignisse, wie das "Magic Event", das "Drachen Event" und das "Tornado Event", bringen eine unvorhersehbare und unterhaltsame Note in das Spiel. So kann das Auto in verschiedenen Szenarien überraschende Fahrmanöver ausführen, was die Spieler herausfordert und das Gesamterlebnis dynamischer gestaltet.

Liste von Fähigkeiten und Features

- Das Auto kann über eine mobile App gesteuert werden
- Modus 1: Live-Ansicht über die Kamera des Autos
- Modus 2: Live-Ansicht über die Handykamera des Benutzers (AR-Modus)
- Magic Event: Das Auto wird angezogen
- Drachen Event: Das Auto fährt rückwärts
- Tornado Event: Das Auto fährt im Kreis
- Die App integriert Hintergrundmusik, um das Spielerlebnis zu verbessern

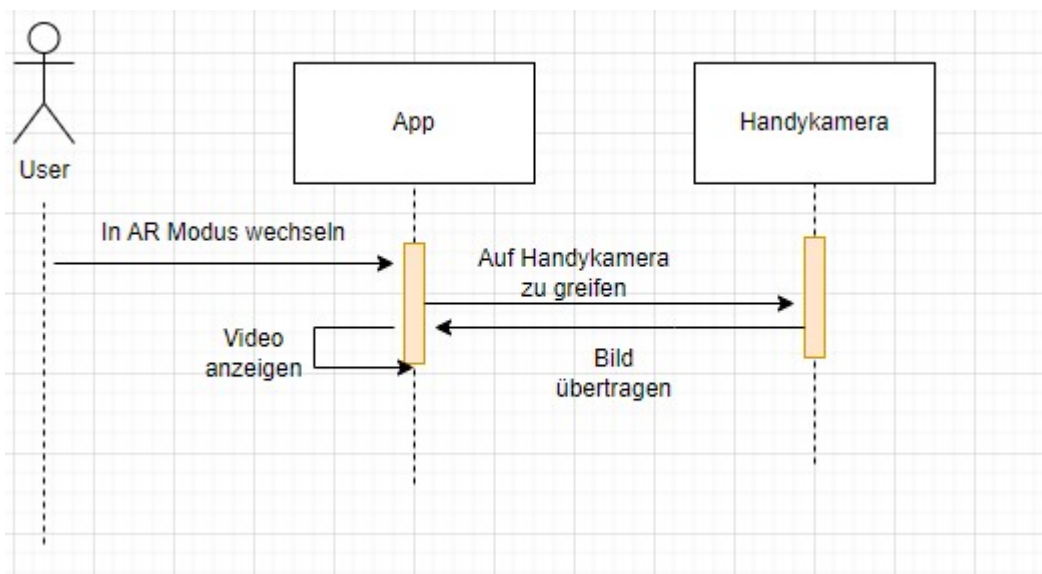
Scenario and Functionality Description

Auto steuern



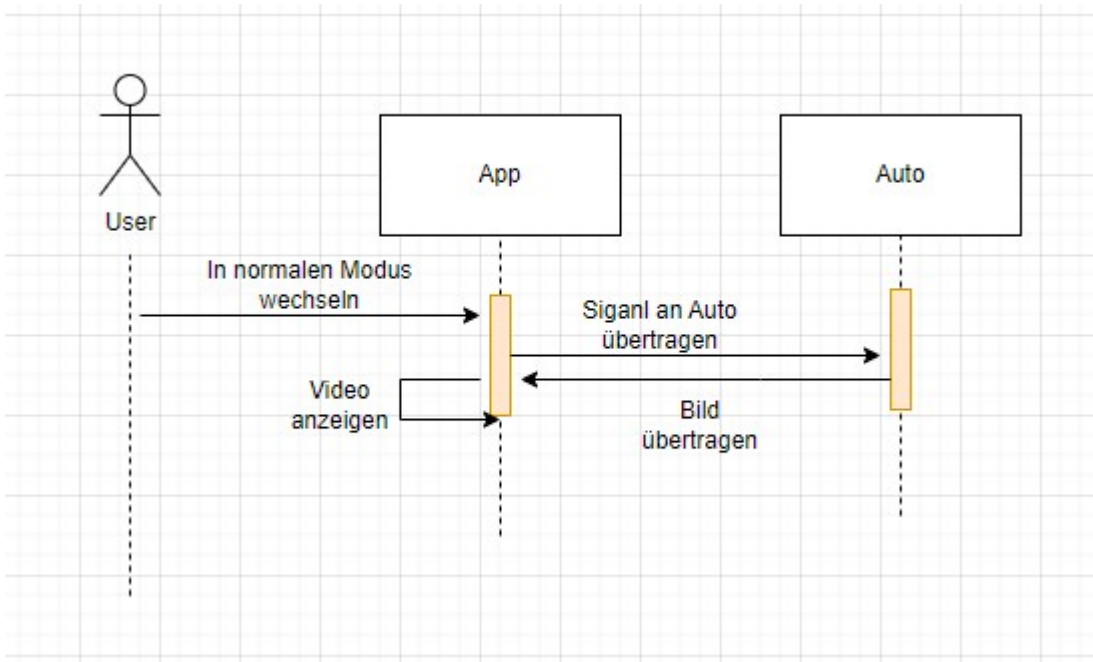
Das Sequenzdiagramm beschreibt den Prozess der Auto-Steuerung, beginnend mit dem Benutzer, der die mobile App verwendet. Der Benutzer initiiert die Steuerung, indem er den Joystick in der App nutzt. Von der App aus wird ein Signal an das Auto übertragen. Das Auto empfängt dieses Signal und führt die entsprechenden Aktionen aus.

Videoübertragung über die Handykamera



Das Sequenzdiagramm illustriert den Prozess der Videoübertragung über die Handykamera im AR-Modus. Der Ablauf beginnt mit dem Benutzer, der in den AR-Modus der App wechselt. Nachdem der Benutzer den AR-Modus aktiviert hat, greift die App auf die Handykamera zu. Die Handykamera überträgt daraufhin das Bild. Die App empfängt das übertragene Bild und zeigt es an.

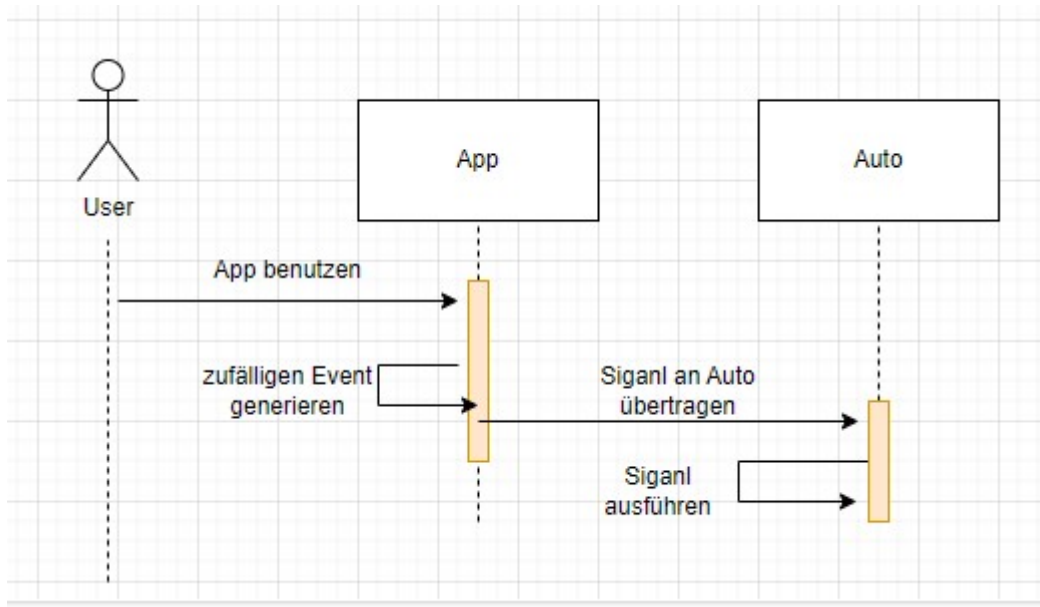
Videoübertragung über die Kamera am Auto



Das Sequenzdiagramm beschreibt den Prozess der Videoübertragung über die Kamera am Auto. Der Ablauf beginnt wiederum mit dem Benutzer, der in den normalen Modus der App wechselt.

Nach der Aktivierung des normalen Modus greift die App nun auf die Kamera am Auto zu. Das Auto überträgt daraufhin das Bild der Kamera. Die App empfängt das übertragene Bild und zeigt es an, ähnlich wie im vorherigen Szenario, jedoch wird hier die Auto-Kamera genutzt.

Events

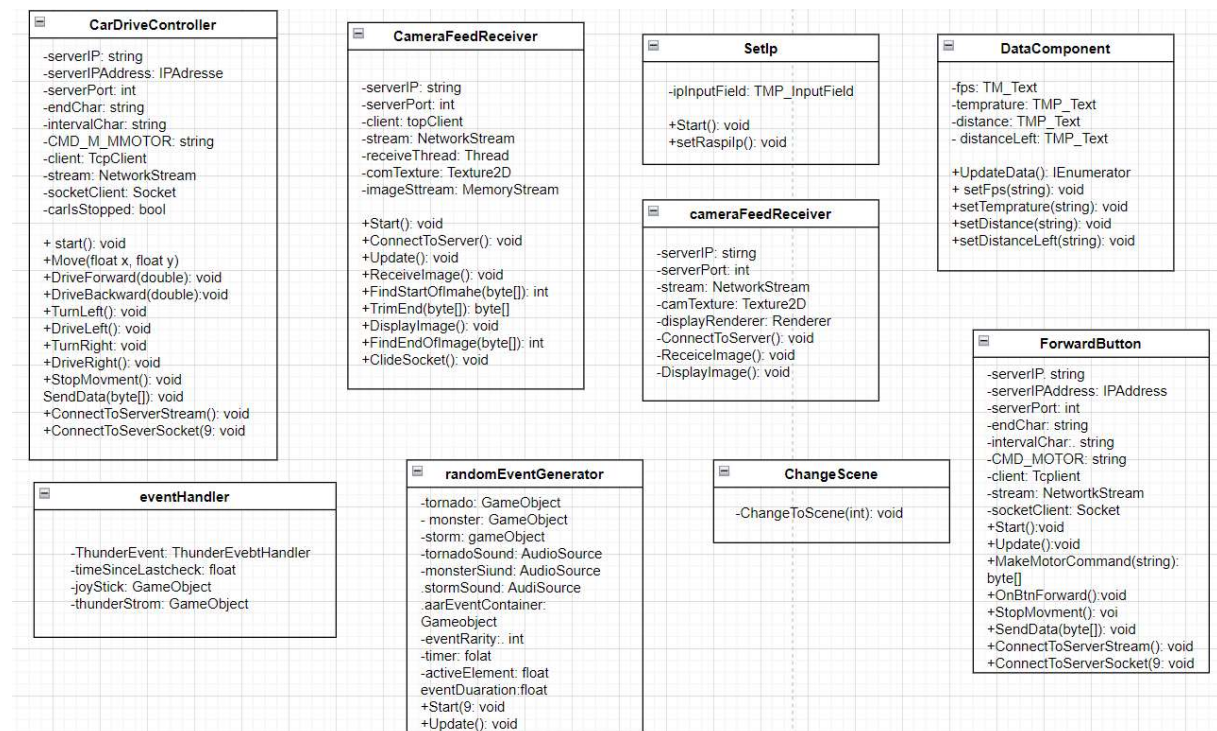


Der sequenzielle Ablauf dieses Szenarios beginnt damit, dass der Benutzer die App verwendet. In diesem Prozess generiert die App eigenständig ein zufälliges Event.

Im nächsten Schritt erfolgt die Übertragung des generierten Events zum Auto. Das Auto, nach Empfang des Signals, führt daraufhin die entsprechende Aktion im Rahmen des zufälligen Events aus. Im AR-Modus der App wird das generierte Event auch visuell dargestellt.

Architecture (internal)

Klassendiagramm



Die Klasse JoystickSetterExample steuert die Bewegung des Autos in Unity. Sie stellt eine Verbindung zu einem Server her und sendet Steuerbefehle für die Motoren des Autos. Die Bewegungsbefehle werden von verschiedenen Methoden wie Move, DriveForward, DriveBackward, TurnLeft, DriveLeft, TurnRight, DriveRight und StopMovement bereitgestellt.

Die Klasse CameraFeedReceiver empfängt Videodaten von einer Kamera über das Netzwerk und zeigt sie in einer Unity-Szene an. Sie stellt eine Verbindung zu einem Server her und verwendet einen Thread, um kontinuierlich Bilder zu empfangen und zu aktualisieren.

Die Klasse ChangeScene ermöglicht das Wechseln zwischen verschiedenen Szenen in Unity. Sie enthält eine Methode ChangeToScene, die aufgerufen wird, um die Szene basierend auf einer übergebenen ID zu ändern.

Die Klasse DataComponent ist für die Anzeige von Daten auf der Benutzeroberfläche verantwortlich. Sie verwendet Coroutine UpdateData, um simulierten Daten wie FPS, Temperatur, Entfernung und verbleibende Entfernung zu aktualisieren.

Die Klasse `randomEventGenerator` erzeugt zufällige Ereignisse, wie einen Tornado, ein Monster oder ein Magic Event, in der Unity-Scene. Die Klasse kümmert sich um die Aktivierung des Events selbst, die Häufigkeit der Events, die Dauer und den Audio- und Visuellen-Output. Dafür nutzt sie einen Zufallszahlengenerator der eine Funktion bei einer bestimmten Zahl triggert. Die Wahrscheinlichkeit für ein Event liegt bei 1 zu 400, um das Fahren sowie das Spielerlebnis nicht zu stark zu beeinträchtigen.

Die Klasse `SetIp` dient dazu, die IP-Adresse für die Verbindung zu einem Server in Unity festzulegen. Sie verwendet Unity's `PlayerPrefs` für die Speicherung und ermöglicht es dem Benutzer, die IP-Adresse über eine UI-Eingabe festzulegen.

Lessons Learned

as a team

Unser Team hat während dieses Projekts eine beeindruckende Lernkurve erlebt, da wir zuvor keinerlei Erfahrung im Bereich Embedded Systems hatten. Der Umgang mit dem Raspberry Pi war für uns alle Neuland, und wir mussten von Grund auf lernen, wie man dieses leistungsfähige Gerät einrichtet und verwendet. Die Herausforderungen, die damit einhergingen, halfen uns, ein tiefes Verständnis für die Funktionsweise von Embedded-Systemen zu entwickeln.

Ein weiterer wichtiger Aspekt unseres Lernprozesses war die Einführung in die Welt der Augmented Reality. Da wir zuvor keine Erfahrung auf diesem Gebiet hatten, war es eine spannende Reise, uns mit den Grundlagen von AR vertraut zu machen. Die Integration der Handykamera in den AR-Modus eröffnete uns neue Perspektiven und zeigte uns die Potenziale dieser aufstrebenden Technologie.

Unity, als Entwicklungsplattform für die App, stellte eine weitere Herausforderung dar, da wir zuvor nur begrenzte Erfahrung damit hatten. Die Auseinandersetzung mit den Funktionen und Möglichkeiten von Unity erforderte eine steile Lernkurve, aber es war eine Erfahrung, die unser Team wachsen ließ.

Insgesamt hat dieses Projekt nicht nur dazu beigetragen, unsere technischen Fähigkeiten zu erweitern, sondern auch unsere Fähigkeit gestärkt, uns in neue

Technologiefelder einzuarbeiten. Die Bewältigung der Herausforderungen hat nicht nur zu einem funktionsfähigen Produkt geführt, sondern auch dazu, dass wir als Team gewachsen sind und nun bereit sind, uns weiteren anspruchsvollen Projekten zu stellen.

Individually

Jenny Jost:

Die Teilnahme an diesem Projekt war für mich persönlich eine herausfordernde und lehrreiche Erfahrung, da ich zuvor keinerlei Erfahrung im Bereich Embedded Systems hatte. Meine bisherigen handwerklichen Fähigkeiten beschränkten sich auf das Zusammenbauen von Möbeln, und die Vorstellung, ein kleines Auto selbst zusammenzuschrauben, war für mich völliges Neuland.

Das Arbeiten mit einem Raspberry Pi war ebenfalls eine Premiere für mich. Die Einrichtung und Nutzung dieses Geräts erforderte intensive Recherche und Einarbeitung in die Grundlagen der Embedded-Systeme. Es war faszinierend zu sehen, wie diese Technologie das Fahrzeug steuerte und welche vielfältigen Möglichkeiten sie bot.

Die Beschäftigung mit dem Thema Batterien stellte sich als komplex heraus, da die Beschaffung nicht so unkompliziert war, wie ich es mir vorgestellt hatte. Die Auseinandersetzung mit verschiedenen Batterietypen, Kapazitäten und deren Integration in das Projekt war eine weitere Lernerfahrung.

Unity war für mich vor diesem Projekt ein weitgehend unbekanntes Terrain. Mit nur begrenzter Erfahrung in der Verwendung dieser Entwicklungsplattform war die Einarbeitung in Unity eine steile Lernkurve. Die Einführung in die Welt der Augmented Reality war eine weitere Herausforderung. Insgesamt hat dieses Projekt meine technischen Fähigkeiten erweitert und mir gezeigt, dass ich in der Lage bin, mich in neue und komplexe Themenbereiche einzuarbeiten. Die Überwindung der anfänglichen Unsicherheiten und die erfolgreiche Umsetzung des Projekts haben nicht nur zu einem funktionsfähigen Produkt geführt, sondern auch mein Selbstvertrauen in Bezug auf technische Herausforderungen gestärkt.

Jakob Lingel:

Während des Verlaufs dieses Projekts habe ich ein Verständnis für die Game Engine Unity entwickelt. Da wir uns entschieden hatten, Virtual-Reality-Elemente in das Projekt zu integrieren, erschien die Wahl dieser Software offensichtlich. Obwohl es im Verlauf der Entwicklung einen Punkt gab, an dem die Verwendung von Unity zweifelhaft erschien, setzten wir fort und schafften es schließlich, einen Weg zu finden, VR-Elemente zu integrieren, indem wir die Kamera des Hostgeräts (Smartphone oder Tablet) statt der Kamera verwendeten, die am Raspberry Pi angeschlossen war und uns viele Probleme bereitet hatte.

Während dieser Arbeit erlangte ich ein gewisses Verständnis für Virtual Reality. Zusätzlich zum Arbeiten mit Unity erlangte ich weiteres Verständnis für den Umgang mit einem SoC wie dem Raspberry Pi. Durch die Implementierung der Fahrzeugsteuerung konnte ich Einblick in die Interaktion von Software mit tatsächlicher Hardware gewinnen. Zuvor hatte ich nie an Software gearbeitet, die Auswirkungen auf die reale Welt hatte. Obwohl ich nicht intensiv am Zusammenbau des Fahrzeugs beteiligt war, war es dennoch aufschlussreich, die Verbindungen der verschiedenen Komponenten zu überprüfen. Um mein Verständnis weiter zu vertiefen, habe ich auch das Handbuch des Fahrzeugs überprüft, um zu sehen, wie alles miteinander zusammen funktioniert.

Lea-Marie:

Das Projekt bot mir einen tiefen Einblick in die Welt der Embedded Systems, insbesondere durch die Steuerung eines Autos mittels Raspberry Pi. Eine bedeutende Komponente dieses Projekts war die Entwicklung von Skripten für den Raspberry Pi, welche die Steuerung der einzelnen Hardwarekomponenten übernahmen. Diese Skripte ermöglichten die nahtlose Integration der Hardware in die Unity-AR-App, indem sie als Schnittstelle zwischen der Software und der physischen Welt fungierten.

Durch die Entwicklung der AR-App in Unity konnte ich wertvolle Erfahrungen sammeln. Da dies meine erste intensive Erfahrung mit Unity und dem zugehörigen AR-Core war, konnte ich diese Kenntnisse nicht nur hier anwenden, sondern auch in Zukunft, da sich meine Bachelorarbeit im Bereich VR/AR bewegen wird.

Während des Projekts stieß ich auf verschiedene Herausforderungen, wie die Integration von Hardware und Software, das Debuggen von Code und die Optimierung der Leistung der AR-Anwendung. Diese Herausforderungen wurden erfolgreich gemeistert, sowohl durch Teamarbeit als auch durch die Unterstützung anderer Gruppenmitglieder.

Insgesamt half mir das Projekt, meine Interessen und Fähigkeiten im Bereich der Embedded Systems weiter zu vertiefen. Die praktischen Erfahrungen, die ich gesammelt habe, haben meine Entscheidung gestärkt, in diesem Bereich als Werkstudent und auch nach meinem Studium tätig zu werden. Die Entwicklung der Skripte für den Raspberry Pi erweiterte mein Verständnis für die Verbindung von Hardware und Software und unterstrich die Bedeutung einer effizienten Kommunikation zwischen den beiden Domänen.

Dieses Projekt war nicht nur eine Lerngelegenheit, sondern auch eine Bestätigung meiner beruflichen Interessen und Ziele. Es hat mir gezeigt, dass Embedded Systems nicht nur faszinierend sind, sondern auch ein Bereich sind, in dem ich meine Fähigkeiten und Leidenschaften erfolgreich einsetzen kann.

Nicolas:

Als jemand, der zuvor noch nie mit Unity oder C# gearbeitet hatte, war dieses Projekt eine intensive, aber ungemein bereichernde Erfahrung. Zu Beginn des Projekts stand das Erlernen der Grundlagen von Unity und C# im Vordergrund. Ich arbeitete mit Szenen, Game Objects und Skripte, lernte, wie man diese Elemente nutzt und miteinander interagiert.

Die Integration von Augmented Reality (AR) mittels Unitys ARFoundation war ein weiterer spannender Aspekt. Ich experimentierte mit dem Platzieren und Interagieren von virtuellen Objekten in einer realen Umgebung, was mein Verständnis für die spannende Welt der AR vertiefte.

Ebenso wichtig war das Event-Handling und das Management von Coroutines, um zeitbasierte Ereignisse zu steuern. Insbesondere das Auslösen von Wettereffekten in zufälligen Intervallen war eine interessante Herausforderung, die meine Fähigkeiten im Umgang mit asynchronen Prozessen in Unity stärkte.

Die Gestaltung der Benutzeroberfläche und die Interaktion mit dem Spieler waren ebenfalls von Bedeutung. Ich erstellte und verwaltete UI-Elemente, die es dem Benutzer ermöglichten, Einstellungen wie IP-Adressen vorzunehmen. Dies gab mir ein besseres Verständnis für die Bedeutung einer benutzerfreundlichen und intuitiven Schnittstelle.

Während des gesamten Projekts war das Debugging und die Problembehandlung eine konstante Begleiterscheinung. Ich lernte, wie man effektiv Fehler identifiziert und löst, was für mich eine der wertvollsten Fähigkeiten in der Softwareentwicklung darstellt. Dadurch konnte der Videostream praktisch auf eine mobile android app übertragen werden.

Zusammenfassend war dieses Projekt nicht nur eine Einführung in Unity und C#, sondern auch eine umfassende Lernerfahrung, die mein Verständnis für Programmierung, Netzwerkkommunikation, AR und Software-Design enorm erweitert hat.

Appendix

19.10.2023 CONOPS MS Report

Project vision description

- A toy car that is controllable per remote (using an app) with a live transmission of the road

Fun Challenge and obstacle courses Scenario:

- Users set up obstacle courses and try to navigate the toy car through them
- Features Needed: Tilt and accelerometer sensors, an app feature to set routes

Indoor adventure Scenario:

- A user wishes to use the car indoor
- Features Needed: car, remote tablet/smartphone

Photography and Filmmaking Scenario:

- Users wish to use the toy car's camera for creative filming or photography purposes
- Features Needed: camera

16.11.2023 Tech Demo MS Report

The car

- includes: camera and sensors
- compatible with Raspberry Pi

Unity app

- Unity AR-App
- Built for Android/iOS

New features

- AR-Opponent for Racing
- Sensors detect obstacles

Communication

- using bluetooth
- maybe using backend
- unity has a library for communication with a raspberry pi

30.11.2023 Architectural Spike MS Report

Milestone Report :

- Auto bekommen + passende Batterien bestellt (Litium Akkus)
- Wireframes für die App erstellt
- Erste Informationen für das Steuern des Autos + deren Funktionen (Kamera Zugriff, Motor, Sensoren, LEDs)
- Informiert über Unity und die Kommunikation mit dem Rasberry Pi

Plan (Milestones) :

Dezember

- Auto zusammensetzen und in Betrieb nehmen + Funktionen kennenlernen und ausprobieren
- Das Steuern über die App mit dem Auto (Kommunikation)
- User Interface für die App als auch die Steuerung
- Objekte zu platzieren und Auto anhand Linie oder vorgegebener Steuerung fahren lassen

Januar

- Umsetzung der AR-Komponente in der App

Herausforderungen :

- Passende Batterien für das Auto zu besorgen –
 - Gelöst: Absprache mit dem Support des Autos und Beratung

21.12.2023 Alpha MS Report

What did we accomplish?

- Auto zusammengebaut, Raspberry Pi OS installiert und isolierte Tests ausgeführt, um die Funktionalitäten zu sichern
- Kommunikation zwischen Raspberry Pi und Unity App
- Videostream von Kamera über TCP
- Anfang der Steuerung über Unity App

What do we plan to do by the next milestone?

- Steuerung des Autos
- Abschluss der UI
- Event-Handling mit obstacles
- Optimierte Kommunikation

What hinderances did we have?

- Servo-Steuerung twitched ab und zu
 - Lösung durch Kommunikation mit Support von Hersteller
- Aufbau der TCP-Verbindung
 - Eigenrecherche

18.01.2024 Beta MS Report

What did we accomplish?

- Bugs gelöst
- Steuerung komplett möglich
- Erste Events eingefügt (Sandsturm, Tornado)
- Kameraübertragung

What do we plan to do by the next milestone?

- Hauptmenü
- Mehr Events und Features (Buzzer & LEDs bei Kollision, Entfernungsmechanik zum nächsten Objekt)
- Events visuell darstellen

- Eventuellen Plattformen-Switch zu React Native

What hinderances did we have?

- Zufälliges ablaufen der Events
- Kamera zuverlässig flüssig wiedergeben
- Anzeigen der Events in der UI