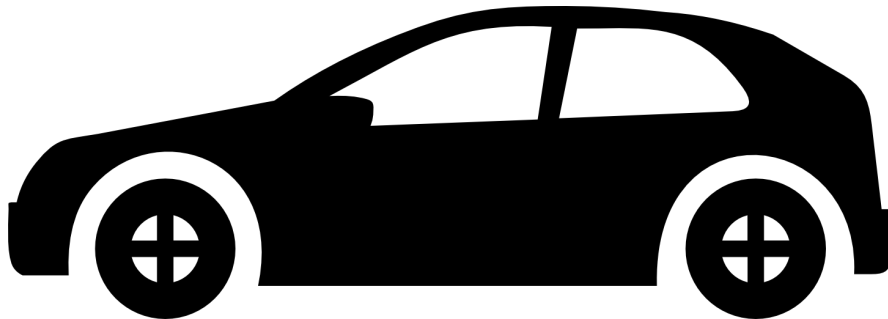


**Hochschule Ruhr West**

Mensch-Technik-Interaktion

Bachelor Projekt: Multimodale Interaktion im Fahrzeug

Sommersemester 2021



# Car Controller

**Vorgelegt von**

Beckmann, Robin      10011874

Thomsen, Felix      10011416

**Abgabe:** 25.09.2021

**Prüfer:** Henrik Detjen

<b>Einleitung</b>	<b>3</b>
<b>Recherche und Ideenfindung</b>	<b>3</b>
<b>Technologie-Stack und Aufbau</b>	<b>5</b>
Server	6
Touchcontroller	6
Voicecontroller	7
<b>Grenzen und Alternativen</b>	<b>8</b>
<b>Kritik</b>	<b>8</b>

# Einleitung

Dies ist die Dokumentation zum Bachelorprojekt "Multimodale Interaktion im Fahrzeug". Im Rahmen dieses Projekts sollte eine Steuerung entwickelt werden, welche es ermöglicht in den Fahrprozess eines autonom fahrenden Autos einzugreifen.

Wir haben uns für dieses Projekt entschieden, da es einen hohen Praxisanteil hat und auf eine fertige Software, beziehungsweise einen fertigen Prototypen hingearbeitet wird.

## Recherche und Ideenfindung

Zu Beginn stellte sich die Frage wie autonom das Fahrzeug fahren kann. In unserem Fall gehen wir von einem Auto aus, welches vollständig autonom fahren kann. Wir wollten dem Nutzer allerdings nicht vollständig die Kontrolle über das Auto entziehen. Es sollen einige Eigenschaften der Fahrt an die Wünsche und Bedürfnisse der Nutzer angepasst werden können. Im ersten Schritt haben wir also überlegt, auf welche Funktionen der Nutzer einen Einfluss haben sollte. Dabei gehen wir davon aus, dass das Auto ohne eingreifen des Nutzers die beste Route und Parameter nutzt, die für den Nutzer und alle anderen Beteiligten am sinnvollsten sind. Der Nutzer kann dann allerdings einige Parameter anpassen, sodass er sich wohl und sicher fühlt. Dazu zählt das Anpassen der Geschwindigkeit, das Wechseln der Spur und Abbiegen sowie Anhalten und ein Nothalt.

Anschließend haben wir überlegt, welche Möglichkeiten zur Eingabe dieser Parameter existieren und welche für uns die Richtigen sind.

Als Möglichkeiten in betracht gezogen haben wir eine Spracheingabe, Kamera gestützte Gestensteuerung, Touch basierte Gestensteuerung, herkömmliche Toucheingaben (so wie bei einem Smartphone oder Tablet), klassische Knöpfe, adaptive Knöpfe (inkl. Knöpfe mit festen Funktionen für kritische Eingaben) und Eye-Tracking.

Nach Abwägung der verschieden Vor- und Nachteile haben wir uns für die Möglichkeit zur Sprach- und Toucheingabe entschieden. Dies hat verschieden Gründe, die wir nun näher erläutern wollen.

Beginnen wir mit der Kamera gestützten Gestensteuerung. Dies ermöglicht es den Nutzern ihre Eingaben frei im Raum zu tätigen, ist quasi blind bedienbar und kann theoretisch zwischen sehr vielen verschiedene Eingaben unterscheiden. Allerdings wird eine Kamera, die die ganze Zeit mit filmt, gerade in Deutschland, von vielen vermutlich nicht gut angenommen. Außerdem ist die rechtslage komplex wie und wann diese Kamera aufnehmen darf und wie diese Bilder verarbeitet werden. Darüber hinaus sind die Gesten nicht unbedingt für alle sofort verständlich und können wenn es viele werden auch leicht verwechselt werden. Auch können Bewegungen verfälscht werden, wenn das Fahrzeug sich

bewegt und steigert somit die Gefahr von Fehleingaben. Auch würde dieser Ansatz einige Nutzer ausschließen, da sie Körperlich nicht in der Lage sind die entsprechenden Gesten auszuführen.

Gesten die auf einer Toucheingabe basieren sind leicht umzusetzen, allerdings abstrakt und müssen daher erst erlernt werden und sind nicht unbedingt für jeden verständlich. Außerdem ist diese Eingabe an einen Ort gebunden und kann ebenfalls durch die Bewegung des Fahrzeuges leicht verfälscht werden.

Gegen klassische Knöpfe haben wir uns entschieden, da diese zwar klar strukturiert sind und momentan von den meisten verstanden und bedient werden können, aber nur sehr wenig Flexibilität bieten. Der Vorteil, dass Knöpfe auch ertastet werden können und ein haptisches Feedback geben, sodass der Fahrer seinen Blick nicht von der Straße wenden muss, wird bei autonom fahrenden Fahrzeugen auch überflüssig.

Mehr Flexibilität hingegen bieten adaptive Knöpfe. Diese können je nach Zustand unterschiedliche Funktionalitäten anbieten. Dadurch kann die Anzahl der Knöpfe reduziert und somit Platz eingespart werden. Allerdings nur im Vergleich zu klassischen Knöpfen. Alle anderen betrachteten Eingabemöglichkeiten nehmen noch weniger Platz ein und bieten eine noch höhere Flexibilität. So können Knöpfe nur eingeschränkt mit Updates und neuen Funktionen versehen werden. Auch könnte es Probleme geben wenn die Nutzer nicht verstehen, in welchem Zustand sich einzelne Knöpfe befinden und warum.

Eye-Tracking kann unserer Meinung nach als unterstützende Technologie genutzt werden, bietet aber kaum oder keine Möglichkeiten als eigenständige Modalität zu arbeiten. Da es eine eher unterbewusste Eingabe ist. So ist es für das Fahrzeug nicht klar, ob bewusst auf etwas geschaut wird oder nur unbewusst, ohne die Intention einer Eingabe.

So gibt es noch die Spracheingabe in Kombination mit einem Touchscreen, der eine Drag-und-Drop Funktionalität ermöglicht. Diese beiden eignen sich unseren Überlegungen nach am besten und ergänzen sich gegenseitig.

Die Spracheingabe ermöglicht es dem Nutzer in einfacher Sprache das Fahrzeug zu steuern. So muss kein neues Konzept erlernt werden, da die Sprache das natürlichste Kommunikationsmittel ist. Dadurch können sehr viele Funktionen abgedeckt werden, da alles relevante in Worte gefasst werden kann. So bietet dies eine natürliche Steuerung die im gesamten Fahrzeug funktioniert und nicht an einen bestimmten Ort gebunden ist.

Rechtlich ist es auch relativ problemlos, da nur das Tonmaterial verarbeitet wird, das unmittelbar nach dem Wakeword gesagt wird und die gesamte Verarbeitung auf dem Gerät stattfindet und keine Daten an andere Unternehmen oder Server sendet. Auch kann das Mikrofon ausgeschaltet werden sollte sich ein Nutzer trotzdem unwohl fühlen. Hier muss

allerdings angemerkt werden, dass stumme Personen keine Möglichkeit der Bedienung haben.

Als zweite Modalität nutzen wir die Drag-und-Drop Toucheingabe. Über einen Touchscreen kann so das echte Fahrzeug gesteuert werden in dem das virtuelle Auto in die Richtung gezogen wird in die das echte Auto sich bewegen soll. Diese Eingabeart soll ebenfalls zur simplen, schnellen Interaktion beitragen.

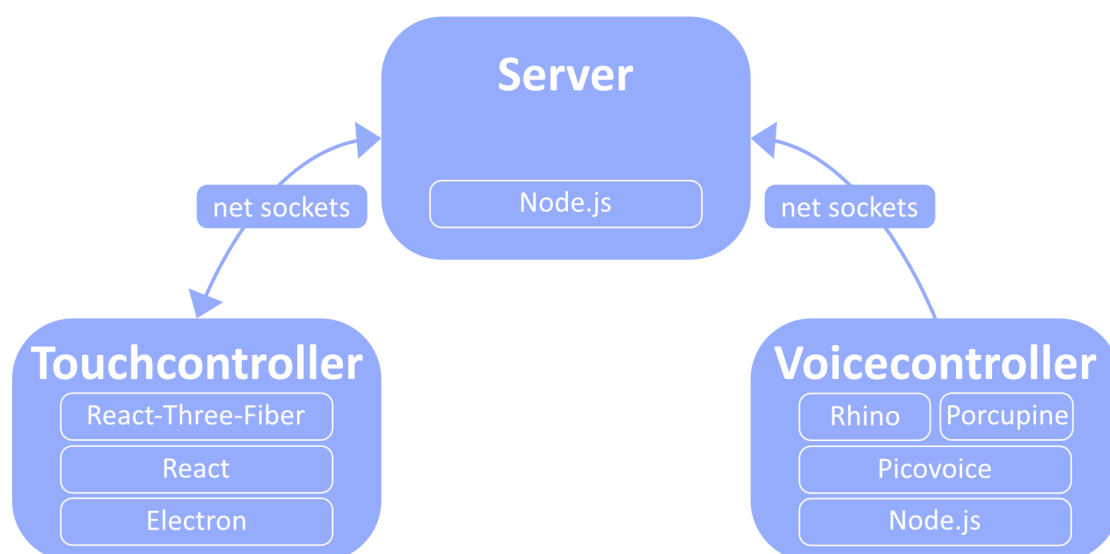
Diese gleicht dann die Probleme der Sprachsteuerung aus und erweitert die Bedienungsmöglichkeit.

## Technologie-Stack und Aufbau

Um dieses Ziel zu erreichen haben wir uns für einen modularen Aufbau entschieden, in dem die Komponenten weitestgehend unabhängig voneinander funktionieren. Dadurch soll es ermöglicht werden, später einfach weitere Komponenten hinzuzufügen oder einzelne Komponenten zu entfernen.

Die Zentrale Rolle spielt dabei der Server, der die Kommunikation mit den einzelnen Eingabegeräten sowie dem Fahrzeug übernimmt.

Als Programmiersprache haben wir uns für Javascript, bzw. Typescript entschieden, da wir diese bereits kennen und sie relativ universell einsetzbar ist. Außerdem gibt es dadurch viele Entwickler, die den Controller in Zukunft potentiell erweitern können. Die Kommunikation läuft über, um json-Sockets erweite, NodeJS net Sockets ab. Im weiteren gehen wir dann auf die Umsetzungen der einzelnen Komponenten ein.



## Server

Das Herzstück bildet der Server. Er kümmert sich um die Kommunikation mit den anderen Komponenten sowie dem Fahrzeug. Umgesetzt als eine NodeJS Applikation, die mittels json-Sockets mit den Eingabegeräten kommuniziert.

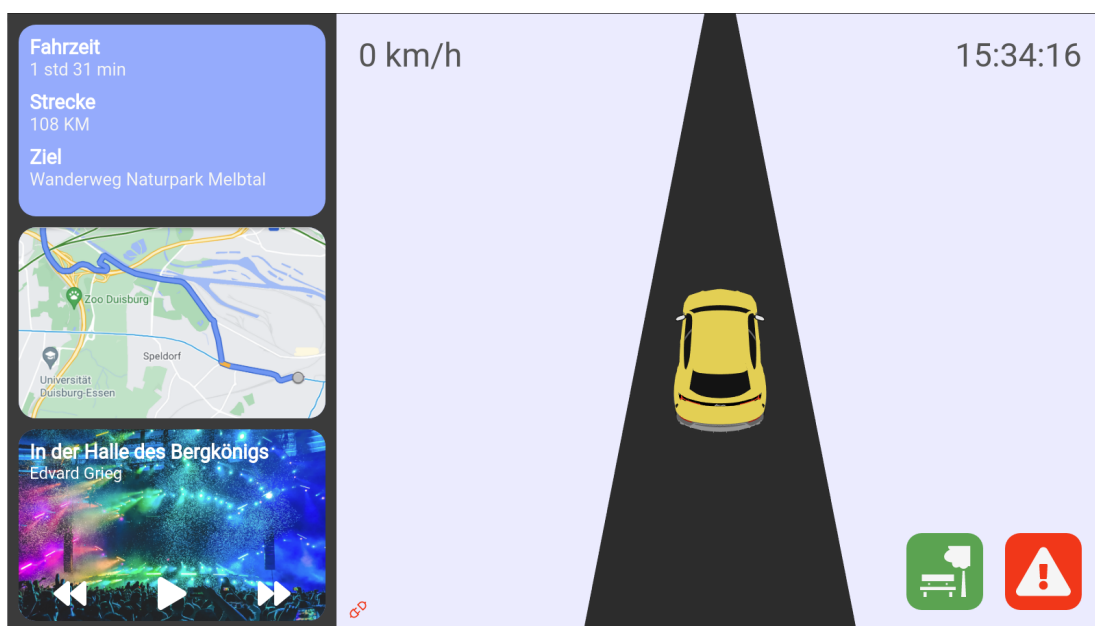
Der Server nimmt die Rolle des Managers ein, der den Zustand des Autos kennt und abfragen kann. Gleichzeitig bekommt er sämtliche Inputs der Eingabegeräte und kann diese dann an das Fahrzeug weitergeben und gegebenenfalls den Status mit anderen Geräten synchronisieren.

## Touchcontroller

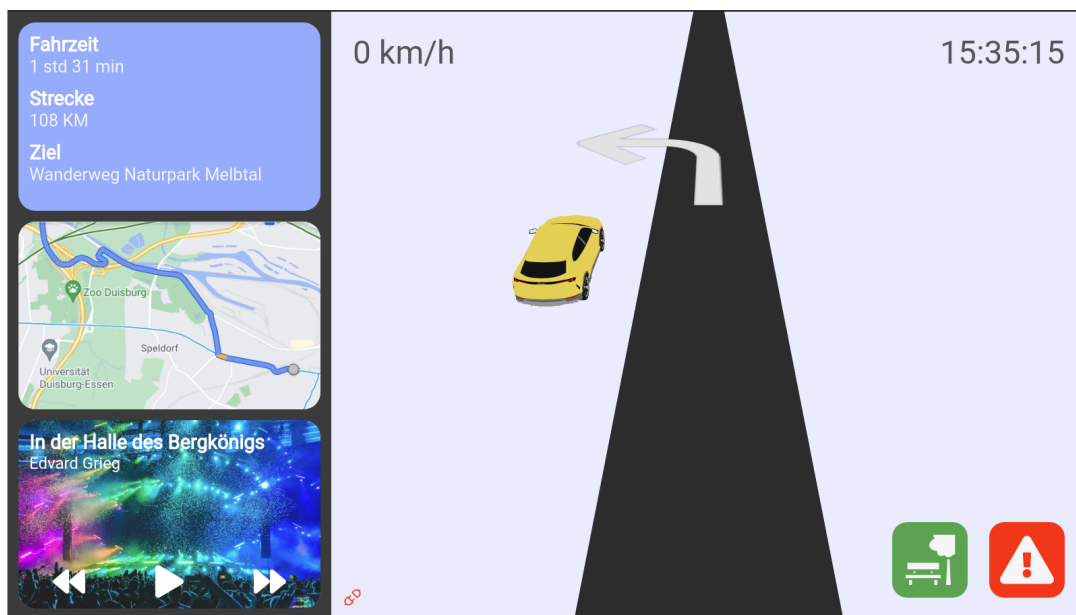
Als Anzeigegerät und gleichzeitig auch Steuereinheit haben wir uns für einen Touchcontroller entschieden. Dieser stellt die aktuelle Situation des Autos dar, in dem Informationen wie Geschwindigkeit, nächste Navigationsschritte oder Ähnliches angezeigt werden. Darüber hinaus können durch die Touchunterstützung auch Eingaben durch die Nutzer getätigt werden, die dann erst an den Server und von diesem dann an das Fahrzeug kommuniziert werden.

Umgesetzt wird dieser ebenfalls mit Typescript und dem Framework electron, welches es erlaubt die Anwendung direkt auf dem Gerät auszuführen und die benötigten Netzwerkschnittstellen zur Verfügung stellt. So erleichtert es die Kommunikation mit dem Server, da so keine Webprotokolle genutzt werden müssen.

Als Frontendframework haben wir uns für React entschieden, welches mit einer 3D Ansicht des Fahrzeuges erweitert wird. Diese wird durch ThreeJS realisiert und mit der passenden React-Three-Fiber library in React integriert.



Der Touchcontroller im Ausgangszustand



Das Auto wird zum abbiegen an die entsprechende Stelle gezogen. Der Prozess wird durch den Indikator auf der Straße visualisiert.

## Voicecontroller

Der Voicecontroller ermöglicht zusätzlich zu der Toucheingabe auch eine Steuerung des Fahrzeuges über die Stimme. Um eine schnelle und vom Internet unabhängige Funktionsfähigkeit zu ermöglichen haben wir uns für Picovoice entschieden, welches aus der Speech-to-Intent Engine Rhino und der Wake Word Engine Porcupine besteht.

Das Wake Word sowie die verschiedenen Intentionen werden dafür im Browser erstellt und dann trainiert, bevor sie mittels des Picovoice SDKs in unsere NodeJS Anwendung eingebunden werden.

Der Voicecontroller nimmt dann über ein Mikrofon die Stimmen auf und wartet, bis das Wake Word **“Hey Auto”** gesagt wird. Das anschließend Gesagte wird dann lokal auf dem Gerät durch Rhino analysiert und verarbeitet. Wenn einer der trainierten Fälle eintritt, wird dieser entsprechend verarbeitet und dann an den Server weitergeleitet.

# Grenzen und Alternativen

Die Spracherkennung durch Picovoice ist gut, allerdings nicht perfekt. Es gibt immer wieder Situationen in denen eine Eingabe nicht erkannt wird und somit keine Aktion an den Server gesendet wird. Außerdem wurden auch einige Eingaben als andere valide Eingaben erkannt. So sollte beispielsweise beschleunigt werden, allerdings wurde das Event des Abbiegens ausgelöst. Dies wird in der Zukunft durch die Weiterentwicklung und Verbesserung von Picovoice vermutlich immer seltener passieren.

Auf Grund dieser Fehler und anfänglicher Recherche nach einer geeigneten Spracherkennung sind wir zu dem Entschluss gekommen, dass dies trotzdem die in dieser Situation beste Lösung ist.

Amazon und andere Tech Unternehmen wie beispielsweise auch Google, bietet zwar eine Schnittstelle für die Nutzung ihrer Spracherkennung an, diese funktioniert allerdings nur mit einer aktiven Internetverbindung welche nicht in jeder Situation im Auto verfügbar ist.

## Kritik

Ein autonomes Auto mit der Fähigkeit die Steuerung gänzlich zu übernehmen kann und wird auf verschiedene Aspekte des Verkehrs optimiert werden. So ist davon auszugehen, dass das Auto zu jeder Zeit die weitsichtigste und beste Entscheidung treffen kann und über Parameter verfügt, die dem Menschen nicht ersichtlich sind. So kann beispielsweise bei einem Überholvorgang die Konsequenz durch einen Computer viel besser evaluiert werden. Denn durch den Vorgang könnten beispielsweise wenige Sekunden gespart werden, jedoch müsste der restliche Verkehr auf das Manöver reagieren und es würde zu einem Stocken kommen der bei allen anderen beteiligten für deutlich mehr Verzögerung führt. Im Idealfall kann man davon ausgehen, dass das Auto besser fährt, als der Mensch und daher ist eine Sinnhaftigkeit der manuellen Steuerung zu hinterfragen. Dennoch verschafft die manuelle Steuerung ein Gefühl der Kontrolle und der Sicherheit und kann zu erhöhter Akzeptanz bei Benutzern führen. Zur Auswirkung wären weitere Untersuchungen sinnvoll.