

# Anki Overdrive

## New Track Editor

Projekt 2 BTI3041

Studiengang:

Autor:

Betreuer:

Datum:

Bsc Informatik

Mac Müller

Prof. Dr. Ing. Reto E. Koenig

21. Januar 2022

# Management-Summary

**In diesem Projekt wurde eine webbasierte Applikation "Track Editor" neu entwickelt, welche das existierende System integriert und ohne Installation funktioniert. Des Weiteren wird ein skalierbares Bildformat in der neuen Applikation eingesetzt und experimentiert.**

## **Ausganglage**

Ein Student der Berner Fachhochschule hat bei seiner Bachelorarbeit ein System für das Rennspiel "Anki Overdrive" entwickelt. Mit seiner Applikation "Track Editor" ist es möglich die Strecke für Fahrzeuge zu erstellen und auszudrucken. Die weitere Funktion "Digital Twin" repräsentiert die Applikation eine virtuelle Darstellung der physikalischen Strecke, worauf sich die Fahrzeuge befinden.

Bevor man das Programm ausführen kann, müssen zusätzliche Software-Pakete mit Kommandozeilen installiert werden, was in der Praxis nicht immer funktioniert hat. In der Softwarearchitektur bestehen zudem viele Kopplungen zwischen Front- und Backend.

## **Zielsetzung**

Das Ziel von dem Projekt "Anki Overdrive" ist eine neue Applikation aufzubauen, welche die "Track Editor" Applikation ersetzen kann. Die Applikation muss neu ohne Backend funktionieren.

## **Umsetzung**

Im ersten Stritt wurde das vorhandene System zusammengebaut. Die Funktionen der alten Applikation wurden aufgelistet. Danach wurden die Anforderungen des Projekts mit dem Steakholder vereinbart.

## **Ergebnisse**

Eine neue Web-Applikation bzw. einen neuen "Track Editor" funktioniert auf der Frontend-Basis. Das Backend wurde nicht mehr benötigt. alle Anforderung wurden erfüllt. Nach einem Experiment wurde festgestellt, dass die Weiterentwicklung des Anki Overdrive System als eine Bachelorthesis möglich wäre.

# Inhaltsverzeichnis

1	Einleitung	4
1.1	Anki Overdrive	4
2	Ausgangslage	5
2.1	Track Editor	6
3	Zielsetzung und Anforderung	8
3.1	[HZ] Hauptziel	8
3.2	[UZ] Unterziele	8
3.2.1	[UZ1] Erstellung der Strecke	8
3.2.2	[UZ2] Digital Twin	8
3.3	Anforderungsarten	8
3.3.1	[FA] Funktionale Anforderungen	9
3.3.2	[NFA] Nicht Funktionale Anforderungen	9
4	Implementation	10
4.1	Design und Modelling	10
4.1.1	Domain Model	10
4.1.2	Systemumfeld	10
4.1.3	Architekturbeschreibungen	10
4.1.4	Systemfunktionalität	11
4.1.5	Nutzer und Zielgruppe	11
4.1.6	Randbedingung	11
4.1.7	Annahmen	11
4.2	Resultat	12
4.2.1	Single Page Web Application (SPA) und MVG-Pattern	13
4.2.2	Deployment Diagram	14
4.2.3	Benutzer Interface	15
4.2.4	Checkliste von den implementierten und nicht implementierten Funktionen	17
5	Schlussfolgerungen/Fazit	18
5.1	Anwendungsanleitung	18
5.2	Bacher-Thesis	18
6	Abbildungsverzeichnis	20
7	Tabellenverzeichnis	20
8	Glossar	21
9	Literaturverzeichnis	21
10	Quellen	21
11	Selbständigkeitserklärung	22

# 1 Einleitung

Im Rahmen des Moduls "BTI3041: Projekt 2" an der Berner Fachhochschule musste eine Bachelor-Thesis weiterentwickelt werden. Aus dieser Thesis entstand ein System für das Rennspiel "Anki Overdrive". Die Applikation "Track Editor", welche diverse Funktionen für den Benutzer zur Verfügung stellt, musste neu entwickelt werden.

## 1.1 Anki Overdrive



Abbildung 1: Anki Overdrive

Anki Overdrive ist ein Rennspiel. Es besteht aus Fahrzeugen (Modell-Auto) und einer Mobile Applikation. Die Rennbahn (Track) kann der Spieler beliebig von verschiedenen Teilen (Track-Piece) zusammenbauen.

Folgende Track-Piece sind online erwerbbar:

- Gerade
- Kurve
- Kreuzung
- Startstück

Als Alternative, geeignet für Softwareentwickler, können die Fahrzeuge über ein Software-Development Kit (SDK) [1] gesteuert werden. Diese übernimmt die Controller Funktion.



Abbildung 2: Track-Piece Code: 1/3 Länge von einer Spur

Unter jedem Track-Piece befindet sich unsichtbare durchgezogene 16 Linien (Spuren). Rechts und links neben der Linie sind drei unterschiedliche Block-Typen, welche den 7 Bit Code wie auf der Abbildung 2 repräsentiert.

Unter jedem Fahrzeug wurde ein Infrarotsensor (IR-Sensor) eingebaut. Somit kann das Fahrzeug einer Linie folgen, den Code ablesen und diesen zum Controller übermitteln.

## 2 Ausganglage

Bei einer Bachelorthesis "Anki Overdrive Track Editor" von Herrn Dominique Marc Hofmann [2] wurde ein System für das Rennspiel "Anki Overdrive" entwickelt.

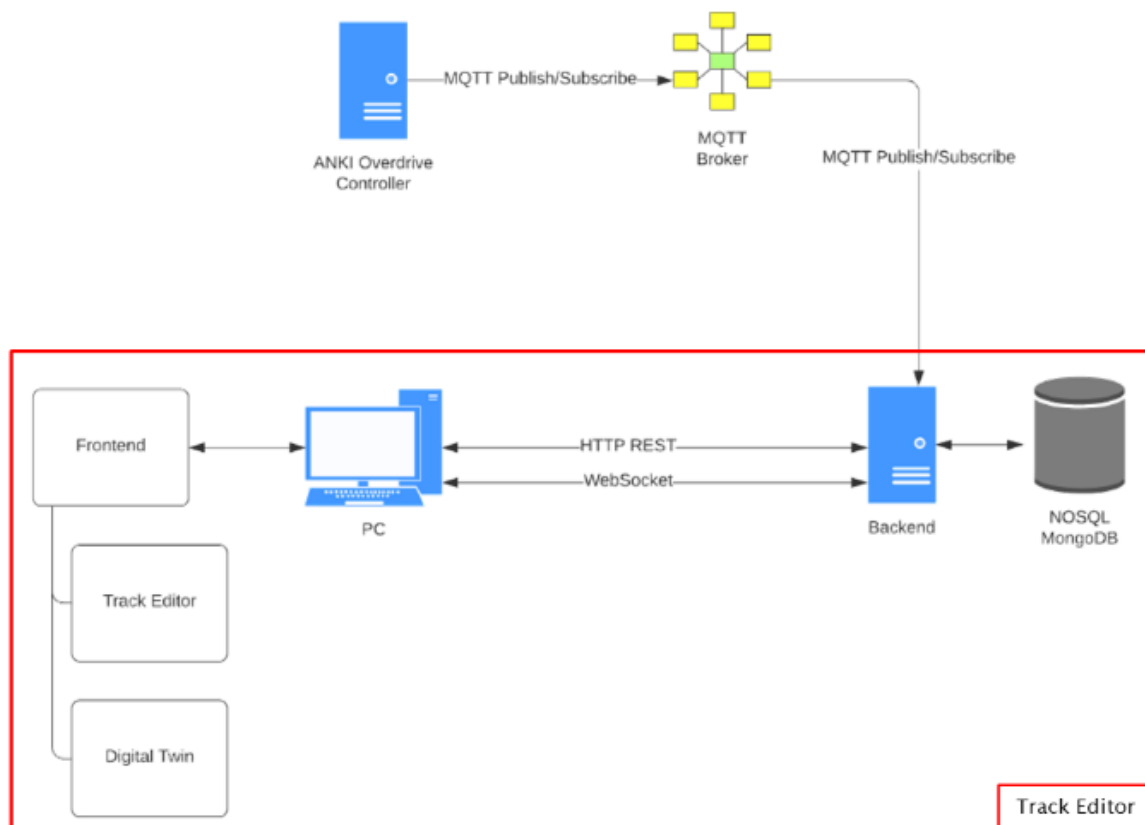


Abbildung 3: Systemarchitektur von der Bachelorthesis

Die Abbildung 3 zeigt, dass die Softwarearchitektur des Track Editors jedoch viele Kopplungen zwischen Front- und Backend aufweist. Das Frontend wurde mit AngularJS-Framework sowie das Backend mit Node.js entwickelt.

Bevor der Benutzer das Programm ausführen kann, müssen zusätzliche Software-Pakete mit Kommandozeilen installiert werden. Das hat in der Praxis nicht reibungslos funktioniert, da der Benutzer eine bestimmte Version der Software bzw. abgekündigte Version installieren muss.

## 2.1 Track Editor

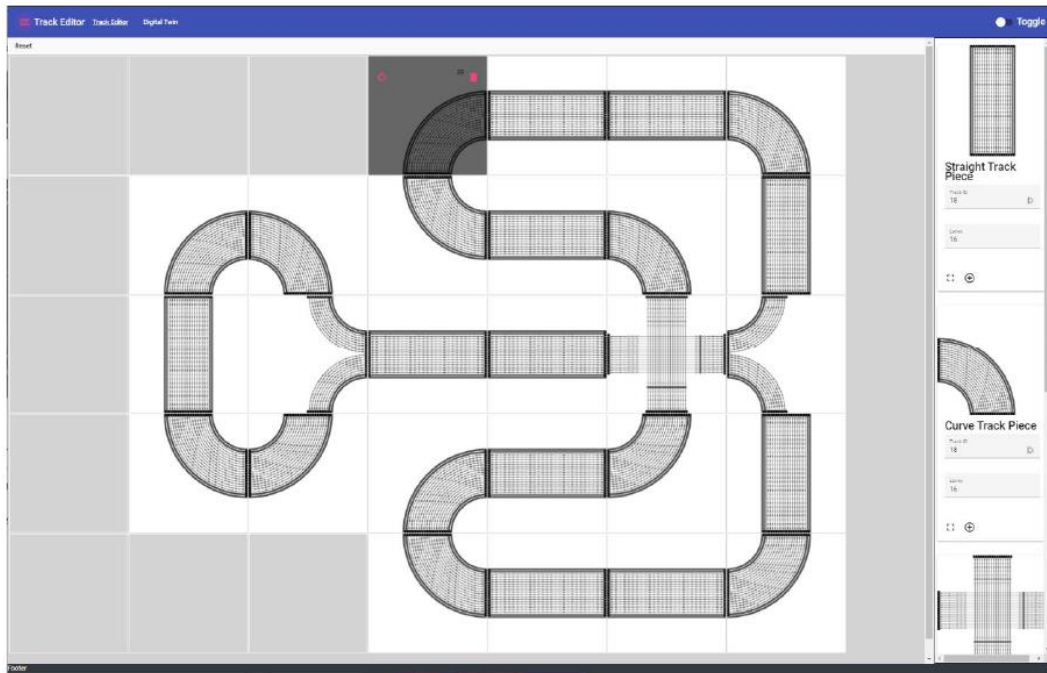


Abbildung 4: Track Editor

Der "Track Editor" ist eine Web-Applikation (Abbildung 4), welche via einen Webbrowser aufrufbar ist. Mit dem "Track Editor" ist es möglich die Strecke für Fahrzeuge selbst zu erstellen und auszudrucken. Eine weitere Funktion "Digital Twin" repräsentiert die Applikation eine virtuelle Darstellung einer physikalischen Strecke, worauf sich die Fahrzeuge befinden.

Die Tabelle 1 enthält eine Auflistung aller Funktionen des Tracks Editors

<b>1 Track Editor</b>
1.1 Editor Board
1.1.1 Funktion pro Track-Piece
1.1.1.1 Rotation
1.1.1.2 Löschen
1.2 Board Kontroller
1.2.1 Zum Board hinzuzufügen: Type
1.2.1.1 Gerade
1.2.1.2 Kurve
1.2.1.3 Kreuzung
1.2.1.4 Verzweigung
1.2.2 Funktion pro Type
1.2.2.1 Eingabe: Track ID Nummer
1.2.2.2 Eingabe: Anzahl der Spuren
1.2.2.3 Ansicht des Kontrollers vergrössern
1.2.2.4 Track-Piece zum Editor Board hinzufügen
<b>2 Einstellungen</b>
2.1 Track Editor
2.1.1 Anpassung der Grösse des Editor Boards
2.1.1.1 Anzahl der Spalte
2.1.1.2 Anzahl der Zeile
2.2 Exportieren
2.2.2 Auswahl
2.2.2.1 Gesamter Track als PNG-Format
2.2.2.2 Alle Track-Piece jeweils als PNG-Format
2.2.2.3 Alle Konfigurationen inkl. generierten Track
2.3 Import
2.3.1 Funktion
2.3.1.1 Konfigurationsdatei importieren
2.4 MQTT
2.4.1 Funktion
2.4.1.1 Eingabe: MQTT-Broker Adresse
<b>3 Digital-Twin</b>
3.1 Digital-Twin Board
3.1.1 Darstellung
3.1.1.1 Track
3.1.1.2 Fahrzeuge
3.2 Fahrzeug Kontroller
3.2.1 Darstellung
3.2.1.1 Kontroller für alle verbundener Fahrzeuge
3.2.1.2 Ansicht Kontroller vergrössern mit mehr Information des Fahrzeugs (Name, Model, Model ID, Product ID)
3.2.2 Funktion
3.2.2.1 Einstellung der Geschwindigkeit
3.2.2.2 Einstellung der Beschleunigung
3.2.2.3 Einstellung der Spuränderung

Tabelle 1: Auflistung aller Funktion des Tracks Editors

## 3 Zielsetzung und Anforderung

### 3.1 [HZ] Hauptziel

**Eine neue Web-Applikation muss die "Track Editor" Applikation ersetzen.**

Das Ziel ist erreicht, wenn die Web-Applikation durch marktübliche Webbrowser beim Benutzer funktioniert. Diese bietet den gleichen Hauptfunktionsumfang wie die alte Version des "Track Editor". Die Backend-Applikation vom alten Track Editor wird vollständig durch die neue Web-Applikation ersetzt.

### 3.2 [UZ] Unterziele

#### 3.2.1 [UZ1] Erstellung der Strecke

**Der Benutzer muss mit der Web-Applikation eine Strecke erstellen können.**

Das Ziel ist erreicht, wenn die Web-Applikation durch marktübliche Webbrowser beim Benutzer funktioniert. Mit dem Webbrowser kann er eine Strecke erstellen und für den Ausdruck exportieren.

#### 3.2.2 [UZ2] Digital Twin

**Der Benutzer kann die virtuelle Darstellung des physikalischen Fahrzeugs auf der Web-Applikation visualisieren lassen.**

Das Ziel ist erreicht, wenn Web-Applikation durch einen gängigen Webbrowser das "Digital Twin" repräsentiert. Ein bereitstehendes Fahrzeug kann verbunden werden.

### 3.3 Anforderungsarten

In diesem Abschnitt werden alle Anforderungen formuliert. Diese wurden dabei in Funktionale und Qualität (Nichtfunktionale) Anforderungen unterteilt.

#### Legende:

In den Tabellen werden folgende Eigenschaften der Anforderungen gebraucht:

**P:** Geschätzte Priorität der Anforderung.

**V:** Variabilität der Anforderung.

**K:** Geschätzte Komplexität der Anforderung.

**R:** Geschätzte Risiko der Anforderung im Zusammenhang mit P, V, K Eigenschaften.

**Quelle:** Von wem kommt diese Anforderung:

Stakeholder (SH), Projektleiter (PL) oder durch eine andere Anforderung.

**Ziele:** Welches Hauptziel bzw. Unterziel werden mit dieser Anforderung erreicht.

Bewertungsgewicht:

↑ = hoch/muss

→ = mittel/soll

↓ = niedrig /tief/"nice to have"



### 3.3.1 [FA] Funktionale Anforderungen

Nr.	Kurzbezeichnung	Beschreibung	P	V	K	R	Quelle	Ziele
FA-01	Erstellung der Strecke	Der Benutzer muss mit der Web-Applikation die Strecke erstellen können. Jeder einzelne Track-Piece muss eine Nummer "Track ID" erhalten.	↑	→	↑	↑	SH	UZ1
FA-02	Strecke Editieren	Der Benutzer kann einen Track-Piece auswählen, anschliessend dieses editieren bzw. löschen.	→	↓	→	→	PL	UZ1
FA-04	Ausdrucken	Der Benutzer soll den einzelnen Track-Piece seines zusammengestellten Tracks als Bilder exportieren und diese ausdrucken können.	→	↓	→	→	PL	UZ1
FA-05	Export-Import	Der Benutzer kann eine Konfiguration seiner zusammengestellten Strecke als Datei exportieren.  Diese Datei kann zu einem späteren Zeitpunkt importiert und weiter editiert werden.	→	→	↓	→	PL	UZ1
FA-06	Digital Twin Mode	Der Benutzer kann den zusammengestellten Track in die Digitale-Zwillings-Ansicht übernehmen und entsprechend visualisieren lassen.	↑	↓	↓	↓	PL	UZ2
FA-07	Live mit MQTT	In der «Digital Twin»-Ansicht können die Fahrzeuge digital abgebildet werden, die sich zu diesem Zeitpunkt auf der realen Strecke befinden bzw. mit dem entsprechenden Kontroller verbunden sind. Der Benutzer kann Befehle über das Web-Interface an die verbundenen Fahrzeuge senden. Die Kommunikation zwischen Web-Applikation und Kontroller funktioniert mit Hilfe des MQTT-Protokolls.	→	→	↓	→	SH	UZ2

Tabelle 2: Funktionale Anforderung

### 3.3.2 [NFA] Nicht Funktionale Anforderungen

Nr.	Kurzbezeichnung	Beschreibung	P	V	K	R	Quelle	Ziele
NFA-01	Entkopplung des Backendes	Die Web-Applikation muss ohne Backend-Software des alten Track Editors weiter funktionieren.	↑	↓	↓	→	SH	HZ
NFA-03	Clean Code	Die Codes der Anwendung sollen nach 4 folgenden Eigenschaften programmiert werden: Lesbarkeit, Änderbarkeit, Erweiterbarkeit, und Wartbarkeit.	↓	↓	↓	→	PL	HZ
NFA-04	Benutzerfreundliches GUI	Die grafische Oberfläche der Anwendung soll benutzerfreundlich sein.	→	↓	→	→	PL	HZ
NFA-05	Sicherheit	Es wird nur eine sichere Verbindung zum MQTT-Broker berücksichtigt. Strengere Sicherheitsmassnahmen können zukünftig implementiert werden.	↓	↑	↑	↑	PL	HZ

Tabelle 3: Nicht Funktionale Anforderung

## 4 Implementation

### 4.1 Design und Modelling

#### 4.1.1 Domain Model

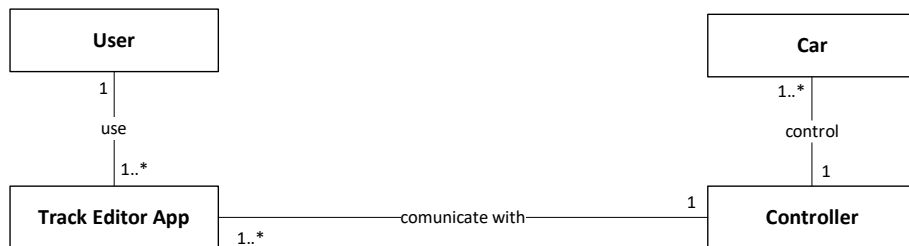


Abbildung 5: Domain Model

Die Abbildung 5 stellt ein Domain Model dar. Ein Benutzer kann ein oder mehrere Track Editor Apps gleichzeitig verwenden. Die Applikationen können jeweils nur mit einem Anki Overdrive Kontroller kommunizieren. Ein Kontroller kann mit mehreren Track Editor App kommunizieren sowie ein oder zahlreiche Fahrzeuge steuern.

#### 4.1.2 Systemumfeld

Das System: Der "Track Editor" soll als eine Web-Applikation dienen. Der Benutzer kann die Strecke für die Fahrzeuge erstellen und die Kommunikation mit den Rennwagen mittels Kontroller herstellen.

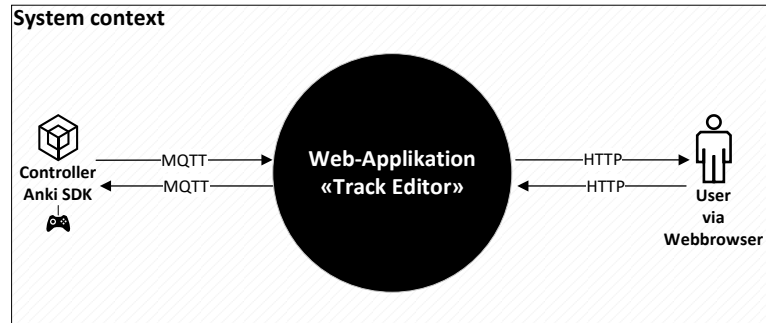


Abbildung 6: Kontextdiagramm

Das Kontextdiagramm auf der Abbildung 6 stellt die Systemkontextabgrenzung sowie den Datenfluss zu dem System dar.

#### 4.1.3 Architekturbeschreibungen

Die Geschäftslogik wird im Frontend als Web-Applikation auf einem Computer ausgeführt. Der Benutzerzugriff erfolgt über einen Webbrowser. Die Applikation kommuniziert mit dem Kontroller über das MQTT-Protokoll.

#### 4.1.4 Systemfunktionalität

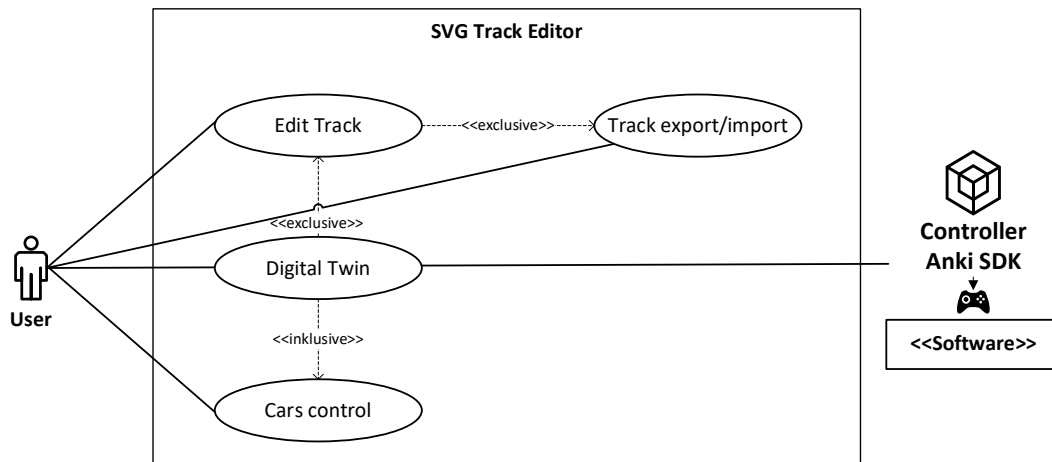


Abbildung 7: Use Case Diagramm

Das Anwendungsfalldiagramm umfasst die vollständigen Anforderungen des Projekts.

#### 4.1.5 Nutzer und Zielgruppe

Die Zielgruppe des Systems sind Personen, welche die Anki Overdrive Fahrzeuge besitzen und die Strecke gerne individuell gestalten möchten.

#### 4.1.6 Randbedingung

Die Benutzer Authentifizierung wird nicht berücksichtigt. Die persistenten Daten werden nicht gespeichert.

#### 4.1.7 Annahmen

Man kann die Anwendung von einem USB-Speicher kopieren, danach auf einen Webserver publizieren und durch einen marktüblichen Web-Browser auf dem PC bedienen. Der Spieler kann die gesamte Strecke ausdrucken. Der grosse Vorteil: Die offiziellen Rennbahnen-Elemente müssen nicht mehr beschafft werden.

## 4.2 Resultat

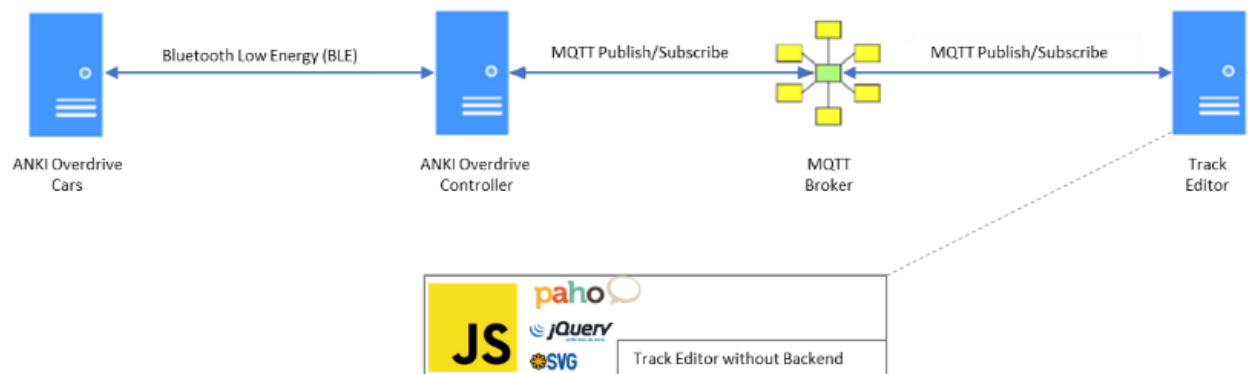


Abbildung 8: Systemarchitektur New Track Editor

Die Abbildung 8 zeigt die neue Systemarchitektur aus diesem Projekt. Der neue Track Editor ohne Backend wurde mit folgenden Technologien eingesetzt:

- JavaScript
- Eclipse paho
- jQuery
- Scalable Vector Graphics (SVG)

Der neuen Track Editor wurde mit JavaScript entwickelt. Die Software-Bibliothek "paho" von der Eclipse Stiftung ermöglicht die Kommunikation zwischen den Track Editor und MQTT Broker über das Websocket [3]. Bei dem Broker wurde einen zusätzlichen Port (TCP 9001) für eine Websocket-Verbindung freigeschaltet [4]. Die jQuery ist ebenso ein Software-Bibliothek. Diese wurde für das "Single Page Application" eingesetzt.

Die skalierbare Vektorgrafik (SVG) wurde von W3C als Standard für Websites empfohlen. Für die Erzeugung des Bildes von Track-Piece stammt der Code von der Bachelor-Thesis, welcher ein Bild mit PNG-Format erzeugt. Ein Track-Piece des neuen Track Editors wurde erst mit SVG-Format erzeugt und dann in ein PNG-Bild Format eingebettet.

#### 4.2.1 Single Page Web Application (SPA) und MVG-Pattern

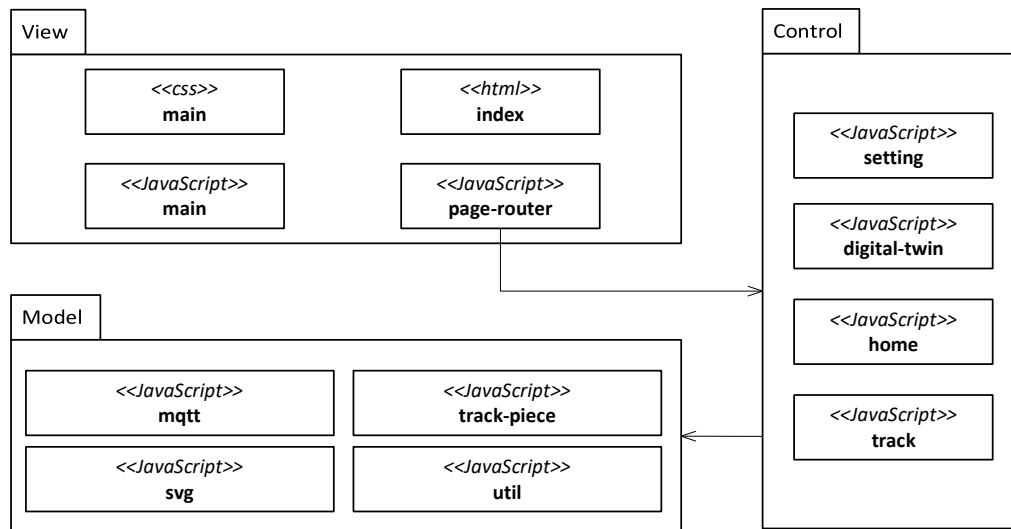


Abbildung 9: SPA und MVG

Für eine benutzerfreundliche Webseite wird eine Single-Page-Webanwendung mit MVC-Pattern verwendet.

Die Beschreibung folgt mit einer Tabelle 4 Unten.

Datei-Name	Beschreibung
index.html	Die Hauptseite der Applikation.
main.css	Die Darstellung von der Applikation wurde in dieser Datei festgelegt.
main.js	Die erste auszuführende JavaScript-Datei. In der Datei wurden alle Pfade zu den Controllern bei page-router.js registriert.
page-router.js	Der Router sucht von dem Benutzer ausgewählte Webpage einen entsprechenden Controller aus. Der Controller generiert die Inhalte der Webseite als HTML-Elemente und gibt diese an den Router zurück. Der Router überschreibt das "<main>"-Element in index.html. Somit muss die Seite bei jeder Anfrage des Benutzers nicht komplett neu geladen werden.
setting.js	Generiert HTML-Elemente und kontrolliert die Funktionen von der "setting"-Seite, welche dem Benutzer den Editor Board und die MQTT-Einstellungen ermöglicht. Die weiteren Funktionen sind Export bzw. Import. Der Benutzer kann den Zustand der Track Editor auf dem Lokal-Computer speichern bzw. wiederherstellen. Der Benutzer ist imstande auch die Tracks als PNG-Bilder für den Drucker herunterladen.
digital-twin.js	Generiert HTML-Elemente und kontrolliert die Funktionen von der "digital-twin"-Seite, welche dem Benutzer die physikalischen Bilder von der Fahrbahn und Fahrzeugen visualisiert. Der Benutzer kann die Geschwindigkeit der verbundenen Fahrzeuge anpassen.
home.js	Generiert HTML-Elemente und kontrolliert die Funktionen von der Hauptseite des Track Editors.
track.js	Ist der Controller für das Board des Track-Editors.
mqtt.js	Ist zuständig für die MQTT-Kommunikation.
track-piece.js	Ist zuständig für die Daten von Track-Piece auf dem Board des Track-Editors.
svg.js	Ist zuständig für die Bearbeitung SVG-Daten vom Track-Piece.
util.js	Ist zuständig für die Session-Daten, Einstellungen, und Import- und Export-Funktionen.

Tabelle 4: Beschreibung SPA & MVG

## 4.2.2 Deployment Diagram

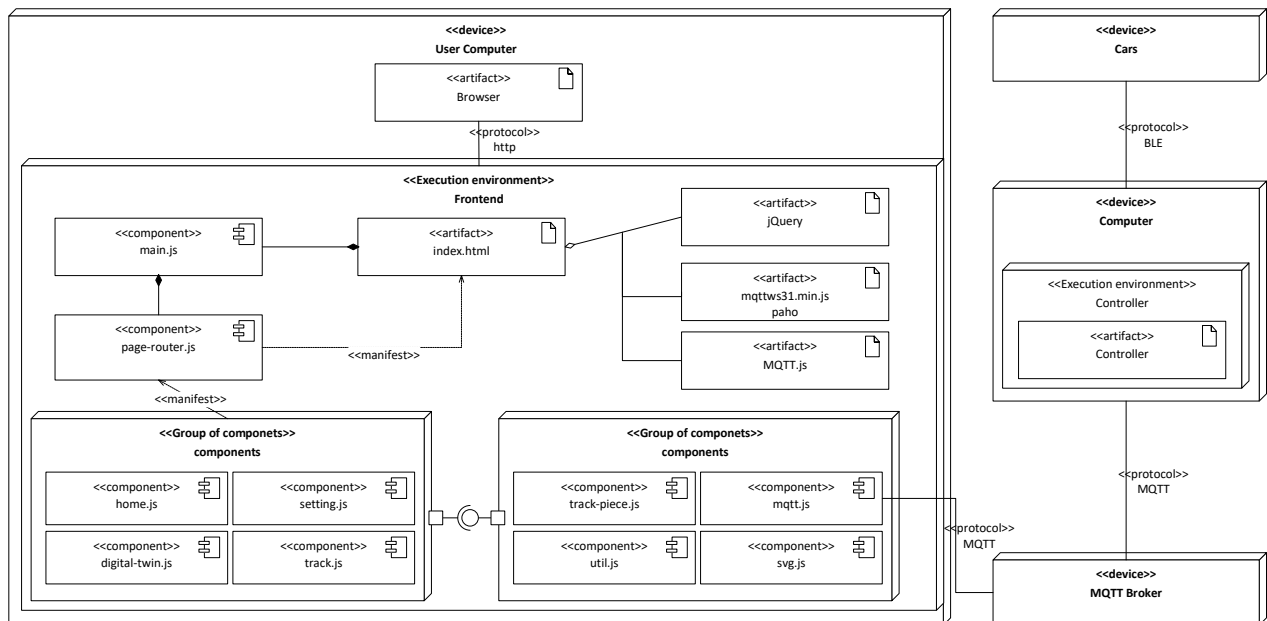


Abbildung 10: Verteilungsdiagramm

Die Abbildung 10 stellt eine Darstellung der Verteilung von Software- und Hardwarekomponenten im Anki Overdrive System.

### 4.2.3 Benutzer Interface

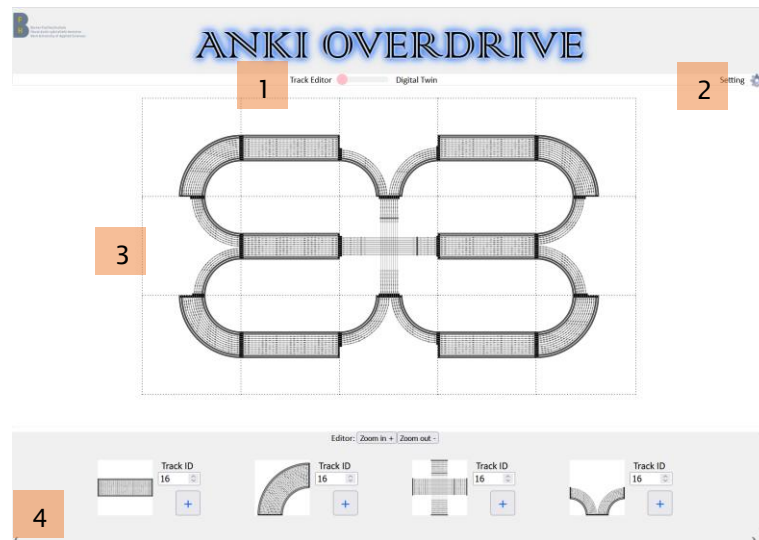


Abbildung 11: Hauptseite: Track Editor

Auf der Hauptseite findet der Benutzer den Track Editor wie auf der Abbildung 11. Die nummerierten Komponenten sind wie folgt:

1. Navigation Switch zwischen Track Editor und Digital Twin
2. Navigation zur Einstellungsseite
3. Das Board des Track Editors
4. Board-Kontroller



Abbildung 12: Digital Twin

Auf der Digital Twin Seite findet der Benutzer die Darstellung wie auf der Abbildung 12. Die nummerierten Komponenten sind wie folgt:

1. Navigation Switch zwischen Track Editor und Digital Twin
2. Navigation zur Einstellungsseite
3. Digital-Twin Board
4. Fahrzeug-Kontroller



Abbildung 13: Setting Seite

Auf der Setting Seite findet der Benutzer die Einstellungsmöglichkeiten wie auf der Abbildung 13. Die nummerierten Komponenten sind wie folgt:

1. Navigation Switch zwischen Track Editor und Digital Twin
2. Navigation zur Einstellungsseite
3. Einstellung für MQTT
4. Einstellung des Editor Boards
5. Testfunktion für die Verbindung zum Anki Overdrive Host und Fahrzeuge
6. Funktionen: Import, Export, Track-Piece als PNG-Bilder auf dem lokalen Computer zu speichern



#### 4.2.4 Checkliste von den implementierten und nicht implementierten Funktionen

<b>1 Track Editor</b>	
<b>1.1 Editor Board</b>	
1.1.1 Funktion pro Track-Piece	
1.1.1.1 Rotation	✓
1.1.1.2 Löschen	✓
<b>1.2 Board Kontroller</b>	
1.2.1 Type	
1.2.1.1 Gerade	✓
1.2.1.2 Kurve	✓
1.2.1.3 Kreuzung	✓
1.2.1.4 Verzweigung	✓
1.2.2 Funktion pro Type	
1.2.2.1 Eingabe: Track ID Nummer	✓
1.2.2.2 Eingabe: Anzahl der Spuren	✗
1.2.2.3 Ansicht des Kontrollers vergrössern	✗
1.2.2.4 Track-Piece zum Editor Board hinzufügen	✓
<b>2 Einstellungen</b>	
<b>2.1 Track Editor</b>	
2.1.1 Grösse des Editor Boards	
2.1.1.1 Anzahl der Spalte	✓
2.1.1.2 Anzahl der Zeile	✓
<b>2.2 Exportieren</b>	
2.2.2 Auswahl	
2.2.2.1 Gesamte Track als PNG-Format	✗
2.2.2.2 Alle Track-Piece jeweils als PNG-Format	✓
2.2.2.3 Alle Konfigurationen inkl. generierten Track	✓
<b>2.3 Import</b>	
2.3.1 Funktion	
2.3.1.1 Konfigurationsdatei importieren	✓
<b>2.4 MQTT</b>	
2.4.1 Funktion	
2.4.1.1 Eingabe: Broker Adresse	✓
<b>3 Digital-Twin</b>	
<b>3.1 Digital-Twin Board</b>	
3.1.1 Darstellung	
3.1.1.1 Track	✓
3.1.1.2 Fahrzeuge	✓
<b>3.2 Fahrzeug Kontroller</b>	
3.2.1 Darstellung	
3.2.1.1 Kontroller für alle verbundener Fahrzeuge	✓
3.2.1.2 Ansicht Kontroller vergrössern mit mehr Information des Fahrzeugs (Name, Model, Model ID, Product ID)	✗
3.2.2 Funktion	
3.2.2.1 Einstellung der Geschwindigkeit	✓
3.2.2.2 Einstellung der Beschleunigung	✗
3.2.2.3 Einstellung der Spuränderung	✗

Tabelle 5: Checkliste von den implementierten und nicht implementierten Funktionen

Die Tabelle 5 enthält die Checkliste von den ✓ implementierten und ✗ nicht implementierten Funktionen.

Die Gründe warum die Funktionen nicht implementiert wurden, folgt bei der nächste Tabelle 6.

Funktion Nr.	Gründe
1.2.2.2	Wenn der Benutzer zu viel Spuren kreiert, könnten die Fahrzeuge die Spuren verwechseln. Deshalb wurde die Anzahl der Spuren auf 16 festgelegt. Dies entspricht auch dem originalen Konzept.
1.2.2.3	Der neuer Board Controller ist auf dem Track Editor gut sichtbar.
2.2.2.1	Ist für den Benutzer nicht notwendig, da es bereits die Funktion 2.2.2.2 gibt. Es kann aber für die Weiterentwicklung implementiert werden.
3.2.1.2	Ist für den Benutzer nicht notwendig. Es kann aber für die Weiterentwicklung implementiert werden.
3.2.2.2	Mit der Beschleunigung wurde nicht experimentiert bzw. wurden keine Information über die Werte gefunden. Die Standard-Einstellung ist bei Wert "5000" festgelegt. Dies hat der Entwickler vom Kontroller ohne Begründung bestätigt.
3.2.2.3	Keine Information über die Werte der Spuränderung. Nach dem vergeblichen Versucht die Werte zwischen 1 bis 10000 einzustellen, wurde festgestellt, dass die Funktion nicht implementiert wurde. Für die Weiterentwicklung ist die Funktion aber wichtig. Der Entwickler wurde bereits kontaktiert.

Tabelle 6: Nicht implementierten Funktion

## 5 Schlussfolgerungen/Fazit

Nach der Implementation entstand eine Web-Applikation bzw. einen neuen "Track Editor". Die neue Applikation funktioniert auf Frontend Basis. Das Backend wurde nicht mehr benötigt. Der Benutzer kann den Track aus verschiedenen Track-Piece zusammenstellen und die Dateien als Bilder mit PNG-Format für den Drucker auf den Computer herunterladen. Die Digital Twin können der physikalische Track und die Fahrzeuge visualisieren. Der Benutzer ist imstande die Geschwindigkeit der Fahrzeuge zu bestimmen.

### 5.1 Anwendungsanleitung

1. Der Anki Overdrive Host als Administrator mit Node.js starten z.B. `$sudo node host.js`
2. Die Station der Fahrzeuge mit Spannung versorgen (Power Adapter einstecken)
3. Die Applikation auf einem Webserver publizieren
4. Die Hauptseite des Webserver aufrufen (index.html)

Bemerkungen:

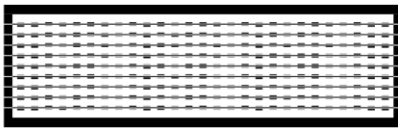
1. Der Benutzer darf die Datei "index.html" nicht direkt vom USB-Stick starten. Die Sicherheitseinstellungen von heutigen Betriebssystemen bzw. Browser beschränken einige Funktionen, wenn sich die Applikation nicht auf dem Webserver befindet
2. Empfohlene Geschwindigkeitswerte sind zwischen 250 bis 750

### 5.2 Bacher-Thesis

Bei diesem Projekt wurde festgestellt, dass die Weiterentwicklung als eine neue Bachelorthesis geeignet wäre. Die Bilder mit PNG-Format hat folgenden Nachteilen:

- Beim Ausdruck wird das Bild verpixelt
- Langsam bei der Erzeugung
- Langsam bei der Bearbeitung
- Grösserer Speicherplatzbedarf

Mit dem skalierbaren SVG Bild-Format, welches XML basiert und heutzutage ein Standard für den Webbrowser ist, würden alle genannten Nachteilen vom PNG-Format abgelöst.



```

<svg width="500" height="500" viewBox="0 0 420 297" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
  <defs>
    <def>
      <path d="M0 5 H 420" stroke="black" fill="transparent" stroke-width="10"/></path>
      <path d="M0 15 H 420" stroke="black" fill="transparent" stroke-width="10" stroke-dasharray="9 402"/></path>
      <path d="M0 20.5 H 411" stroke="black" fill="transparent" stroke-width="1"/></path>
      <path d="M14 18.9 H 486" stroke="black" fill="transparent" stroke-width="0.4" stroke-dasharray="7.39622641509434"/></path>
      <path d="M14 22.1 H 486" stroke="black" fill="transparent" stroke-width="0.4" stroke-dasharray="7.39622641509434"/></path>
      <path d="M14 22.4 H 486" stroke="black" fill="transparent" stroke-width="1.6" stroke-dasharray="7.39622641509434, 7.39622641509434, 110.9434, 110.9433962264151, 7.39622641509434, 110.9433962264151"/></path>
      <path d="M14 22.1 H 486" stroke="black" fill="transparent" stroke-width="1" stroke-dasharray="0, 44.37735849656604, 7.39622641509434, 22.18867924528382, 7.39622641509434, 7.39622641509434, 1000000000"/></path>
      <path d="M14 18.9 H 486" stroke="black" fill="transparent" stroke-width="1" stroke-dasharray="0, 29.58490566037736, 7.39622641509434, 22.18867924528382, 7.39622641509434, 36.0811320754717, 7.39622641509434, 1000000000"/></path>
      <path d="M0 26 H 420" stroke="black" fill="transparent" stroke-width="10" stroke-dasharray="9 402"/></path>
      <path d="M0 31.5 H 411" stroke="black" fill="transparent" stroke-width="1"/></path>
      <path d="M14 29.9 H 486" stroke="black" fill="transparent" stroke-width="0.4" stroke-dasharray="7.39622641509434"/></path>
      <path d="M14 33.1 H 486" stroke="black" fill="transparent" stroke-width="0.4" stroke-dasharray="7.39622641509434"/></path>
      <path d="M14 33.4 H 486" stroke="black" fill="transparent" stroke-width="1.6" stroke-dasharray="7.39622641509434, 7.39622641509434, 110.9434, 110.9433962264151, 7.39622641509434, 110.9433962264151"/></path>
    </def>
  </defs>
</svg>

```

Abbildung 14: Gerade Track-Piece mit SVG-Format

Abbildung 13 links visualisiert eine erzeugte Track-Piece gerade mit dem SVG-Format. Rechts ersichtlich ist ein Abschnitt vom Code der SVG Datei. Die Erzeugungsmethoden sind komplex und deshalb geeignet für eine tiefere Untersuchungen mit einer Bachelorthesis.

Ein weiterer Punkt ist die Weiterentwicklung mit Anki Overdrive System. Das System könnte viel einfacher für Benutzer sein. Der Anki Overdrive Host, MQTT-Server, und Track Editor könnten sich zusammen auf einem kleinen Computer befinden und automatisch nach Inbetriebnahme funktionieren. Der Benutzer könnte den Track Editor direkt mit dem Browser über die IP-Adresse aufrufen.

## 6 Abbildungsverzeichnis

Abbildung 1: Anki Overdrive	4
Abbildung 2: Track-Piece Code: 1/3 Länge von einer Spur	4
Abbildung 3: Systemarchitektur von der Bachelorthesis	5
Abbildung 4: Track Editor	6
Abbildung 5: Domain Model	10
Abbildung 6: Kontextdiagramm	10
Abbildung 7: Use Case Diagramm	11
Abbildung 8: Systemarchitektur New Track Editor	12
Abbildung 9: SPA und MVG	13
Abbildung 10: Verteilungsdiagramm	14
Abbildung 11: Hauptseite: Track Editor	15
Abbildung 12: Digital Twin	15
Abbildung 13: Setting Seite	16
Abbildung 14: Gerade Track-Piece mit SVG-Format	19

## 7 Tabellenverzeichnis

Tabelle 1:Auflistung aller Funktion des Tracks Editors	7
Tabelle 2: Funktionale Anforderung	9
Tabelle 3: Nicht Funktionale Anforderung	9
Tabelle 4: Beschreibung SPA & MVG	13
Tabelle 5: Checkliste von den implementierten und nicht implementierten Funktionen	17
Tabelle 6: Nicht implementierten Funktion	18

## 8 Glossar

Begriff	Beschreibung
Software Development Kit (SDK)	Eine Software, die bei der Entwicklung hilft. Die Softwareentwickler können die verfügbare Dienste nutzen.
Framework	Eine Vorprogrammierung, welches die verschiedenen Funktionen zur Verfügung stellt.
AngularJS	Ein clientseitiges JavaScript-Webframework zur Erstellung von Webseiten.
Node.js	Ist eine plattformübergreifende JavaScript-Laufzeitumgebung.
Frontend	Schichteneinteilung bei der Softwareentwicklung: Nutzeroberfläche
Backend	Schichteneinteilung bei der Softwareentwicklung: Teil eines Systems, dass sich mit der Datenverarbeitung im Hintergrund beschäftigt.
Digital Twin	Ein digitaler Zwilling ist eine digitale Repräsentanz eines materiellen oder immateriellen Objekts aus der realen Welt in der digitalen Welt.[4]
MVC-Pattern	Model View Controller (MVC): Ein Architekturmuster für die Softwareentwicklung.
IP-Adresse	Internetprotokoll-Adresse: Eine eindeutige Identifizierung und Adressierung eines Geräts im Netzwerk.
W3C	The World Wide Web Consortium (W3C)

## 9 Literaturverzeichnis

[1] Anki, Inc. anki/drive-sdk. Zugriff am: 21.01.2021. <a href="https://github.com/anki/drive-sdk">https://github.com/anki/drive-sdk</a>	4
[2] Hofmann, Dominique Marc. Bachelor-Thesis: Anki Overdrive Track Editor. 16.06.2021	5
[3] Cope, Steve. Using MQTT Over WebSockets with Mosquitto. Zugriff am: 21.01.2021. <a href="http://www.steves-internet-guide.com/mqtt-websockets/">http://www.steves-internet-guide.com/mqtt-websockets/</a>	12
[4] Cope, Steve. Configuring Websockets On Your Own Mosquitto Broker Zugriff am: 21.01.2021. <a href="http://www.steves-internet-guide.com/mqtt-websockets/">http://www.steves-internet-guide.com/mqtt-websockets/</a>	12
[4] Wikipedia. Zugriff am: 21.01.2021. <a href="https://de.wikipedia.org/wiki/Digitaler_Zwilling">https://de.wikipedia.org/wiki/Digitaler_Zwilling</a>	20

## 10 Quellen

Abbildung 1: [https://m.media-amazon.com/images/I/71laq5Aa7uL.\\_SL1500\\_.jpg](https://m.media-amazon.com/images/I/71laq5Aa7uL._SL1500_.jpg)

Abbildung 2,3,4: Bachelor-Thesis: Anki Overdrive Track Editor, 16.06.2021

## 11 Selbständigkeitserklärung

Ich bestätige, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der im Literaturverzeichnis angegebenen Quellen und Hilfsmittel angefertigt habe. Sämtliche Textstellen, die nicht von mir stammen, sind als Zitate gekennzeichnet und mit dem genauen Hinweis auf ihre Herkunft versehen.

Ort, Datum: Luzern, 21.01.2022

Unterschrift: Mac Müller