

Software Engineering Projekt

Crazy Car

Author...

23. April 2020

Inhaltsverzeichnis

1	Vision	3
2	Epic und User Stories	3
2.1	Epic	3
2.2	User Stories	3
3	Kamera	5
4	Schnittstelle zur Steuerung	5
5	Anmerkung in Bezug auf Covid-19 Situation	7
6	Simulator	8

1 Vision

Nach Projektabschluss ist das CrazyCar über ein Programm am PC mithilfe der Pfeiltasten der Tastatur fernsteuerbar.

Unterstützend wird am PC auch ein Videostream der am Fahrzeug montierten Kamera angezeigt, sodass das Fahrzeug mithilfe des Kamerabildes gesteuert werden kann. Die Übertragung der Daten zwischen dem Fahrzeug und dem PC erfolgt via WLAN.

2 Epic und User Stories

Im folgenden Teil werden User Stories sowie der Epic unseres Projekts dargelegt. Dabei sollen diese die wesentlichen Anforderungen an das Softwareprojekt festlegen, um den Nutzern einen optimalen Funktionsumfang zu bieten.

2.1 Epic

Nr. 1	Remote Control CrazyCar
Das sichere Steuern des CrazyCars ohne Sichtkontakt.	

2.2 User Stories

Nr. 1-1	First Person Camera
Als Benutzer bzw. Student möchte ich eine verzögerungsfreie Bildübertragung damit ich dank der Kamera mögliche Hindernisse sehe und darauf in meinem Fahrverhalten reagieren kann.	

Nr. 1-2

User Friendly

Als Benutzer (Student, Kind)

möchte ich ein einfaches User Interface

damit ich das CrazyCar einfach steuern kann, keine Commandline Befehle notwendig sind und ich mich ohne Fachwissen ausstoben kann.

Nr. 1-3

Safety First

Als zahlender Besitzer des Autos (Lehrstuhl – CIT)

möchte ich einen Notstop

damit vor Hindernissen rechtzeitig gestoppt wird und damit in weiterer Folge das Auto nicht gleich kaputt ist.

3 Kamera

In der ersten Ausbaustufe des Projekts wird als Kamera ein Smartphone verwendet, welches durch eine App die Bildübertragung ermöglicht. Nach Recherchen fiel die Wahl auf die für Android und iOS verfügbare App *Instant Webcam*, welche kostenfrei in den jeweiligen App Stores geladen werden kann. Über die App kann dann in verschiedenen Qualitäten eine Bildübertragung erfolgen, was in der niedrigsten Qualität verzögerungsfrei funktioniert. Der Videostream ist dann über eine URL in jedem Browser abrufbar und ist im jeweiligen WLAN Netzwerk verfügbar. In Abbildung 1 ist die Benutzeroberfläche der App dargestellt auf welcher auch die entsprechende URL zum Abrufen des Video Streams ersichtlich ist.

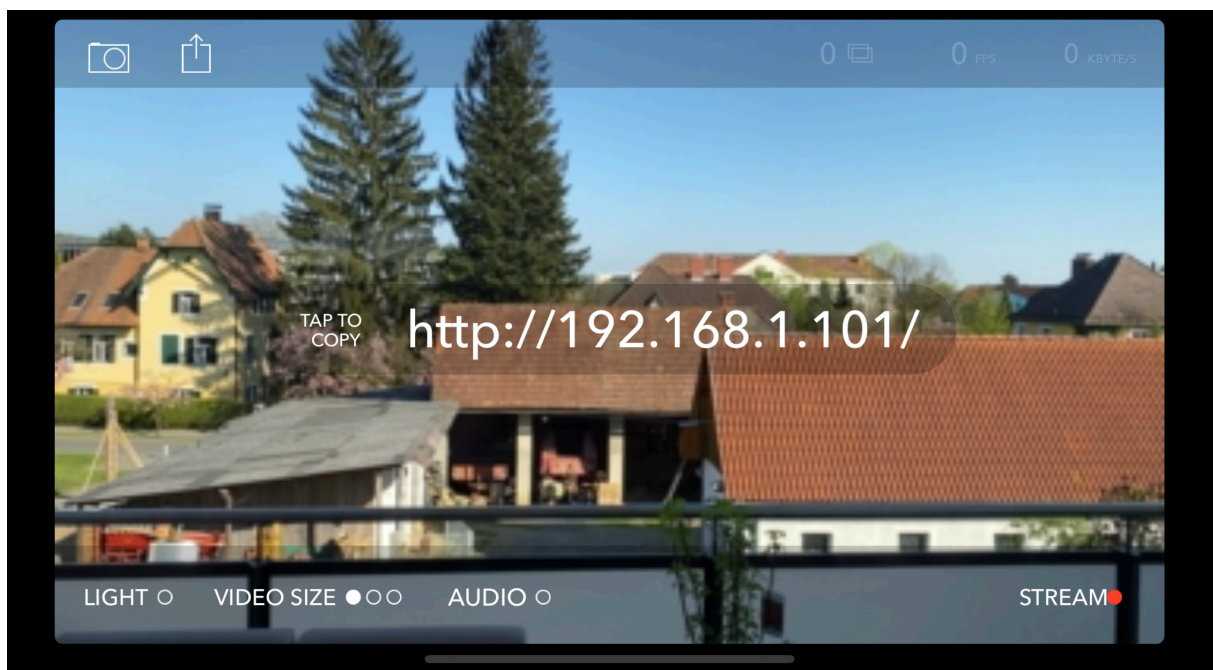


Abbildung 1: Oberfläche von Instant Webcam

Im unteren Teil des UI ist wie bereits erwähnt die Qualität der Übertragung einstellbar. Die niedrigste Qualitätsstufe ist für die Zwecke des CrazyCars ausreichend.

4 Schnittstelle zur Steuerung

Vom Auftraggeber wurde ein Java Interface mit den Funktionen des CrazyCars zur Verfügung gestellt, auf dessen Basis die Steuerungslogik sowie der Nothalt implementiert wird.

Die im Interface definierten Methoden werden in der Klasse *CarImpl* und der Nothalt sowie andere unterstützende Funktionen in der Klasse *DriveAssistant* implementiert.

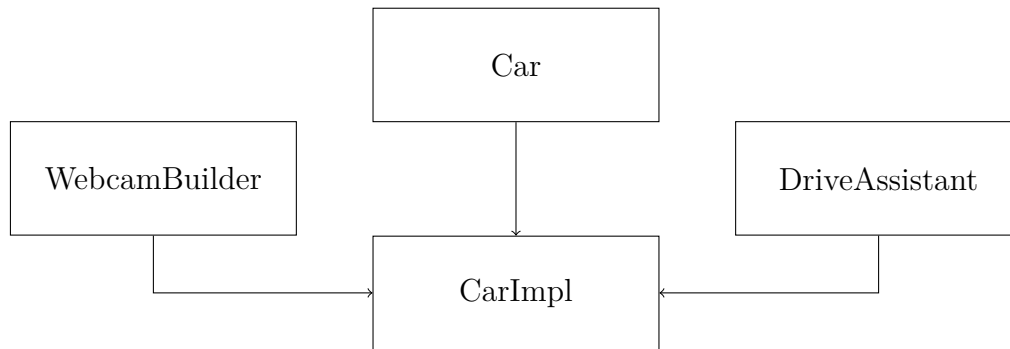


Abbildung 2: Klassen im Projekt

Wie bereits erwähnt ist die Klasse *DriveAssistant* für eine Unterstützung des Fahrers zuständig und stellt die Nothalt Funktion bereit. Die Klasse *WebcamBuilder* ist wiederum für den Aufbau der Verbindung mit der Webcam verantwortlich und stellt sicher, dass eine stabile Verbindung mit der Kamera aufgebaut wurde. Der Nutzer wird dabei wie in Abbildung 3 aufgefordert, die in der App auf dem Smartphone ersichtliche URL einzugeben.

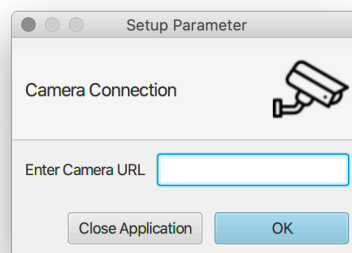


Abbildung 3: Aufforderung zur Eingabe der Kamera URL

Hat der Verbindungsaufbau geklappt, wird der Nutzer weitergeleitet. Im Fehlerfall (Textfeld leer, Kamera nicht erreichbar,...) wird der Nutzer aufgefordert seine Eingabe oder die Verbindung der Kamera mit dem WLAN zu überprüfen.

Nach erfolgreichem Verbindungsaufbau wird der Nutzer zu einem minimalistisch gehalten User Interface weitergeleitet. Dieses ist in Abbildung 4 dargestellt und teilt sich in drei Komponenten.

- Camera Section
- Controller Section
- Status Section

Im Bereich *Camera Section* wird das aktuelle Bild der Kamera angezeigt, mit der zuvor eine Verbindung aufgebaut wurde. Die *Controller Section* stellt die notwendigsten Steuerelemente zur Verfügung, um Fahrbefehle an das CrazyCar zu übermitteln. In der *Status Section* werden aktuelle Werte (z.B. Geschwindigkeit, Messwerte der Distanzsensoren, etc.) dargestellt.

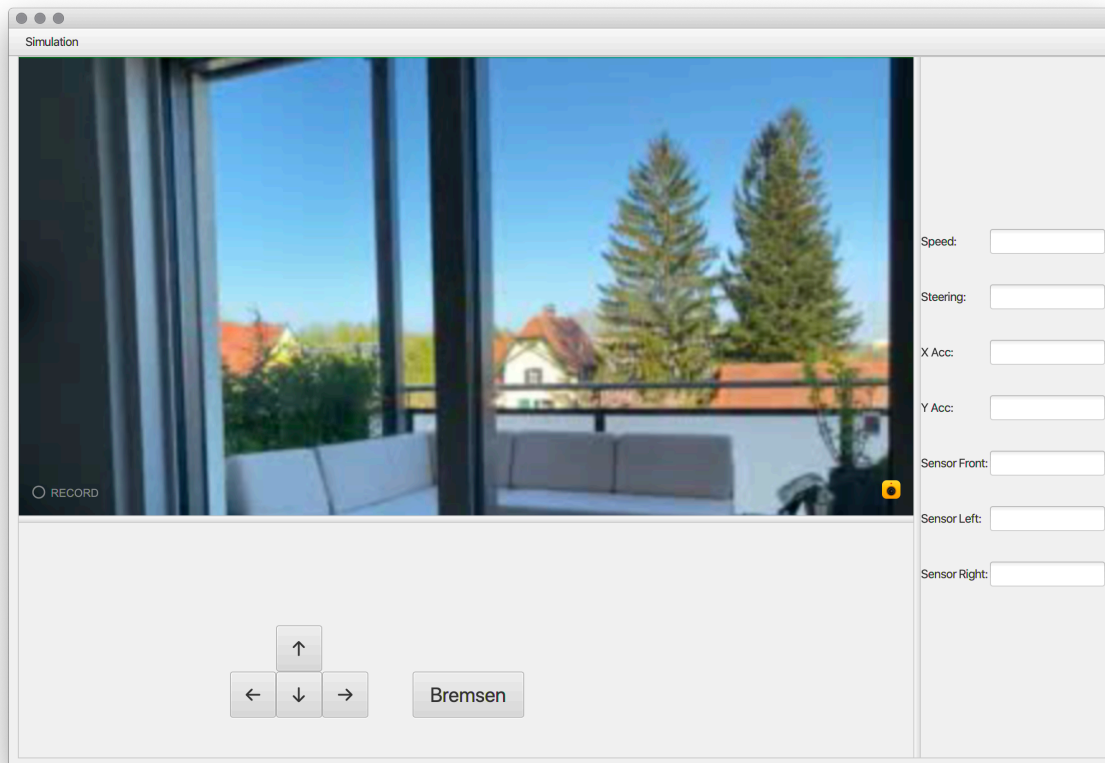


Abbildung 4: User Interface Hauptbildschirm

5 Anmerkung in Bezug auf Covid-19 Situation

Die CrazyCar MUL Software Development GmbH trägt selbstverständlich die von der österreichischen Bundesregierung verordneten Maßnahmen zur Eindämmung der Covid-19 Pandemie mit - nicht zuletzt zum Schutz unserer Kunden und Mitarbeiter.

Doch ist es unser Bestreben Projekte trotz dieser Schwierigen Umstände fristgerecht abwickeln zu können. Aus diesem Grund haben wir vorab einen Simulator in das Projekt integriert, um Ihnen und Ihrem Team die Möglichkeit zu geben die Funktionalitäten zu testen. Dabei wird im virtuellen CrazyCar das gleiche Interface implementiert wie es auch in der echten Version der Fall ist. Der Vorteil dieser Vorgehensweise ist die einfache Umstellung auf Ihr reales Crazy Car, da wir mit Ihnen gemeinsam die Testphase in virtueller Umgebung abwickeln können.

6 Simulator

Der Simulator kann über den Menüpunkt *Simulation*→*New Simulation Window* geöffnet werden. Das Fenster das sich dann öffnet ist Ihre virtuelle Welt.

Im oberen Bereich werden aktuelle Messwerte/Daten des CrazyCars angezeigt. Durch den Reset Knopf können alle in der Welt befindlichen Hindernisse entfernt werden. Die Steuerung des CrazyCars erfolgt über die *Control Section* im Hauptbildschirm und in der *Status Section* werden ebenfalls die aktuellen Werte/Daten des CrazyCars angezeigt.

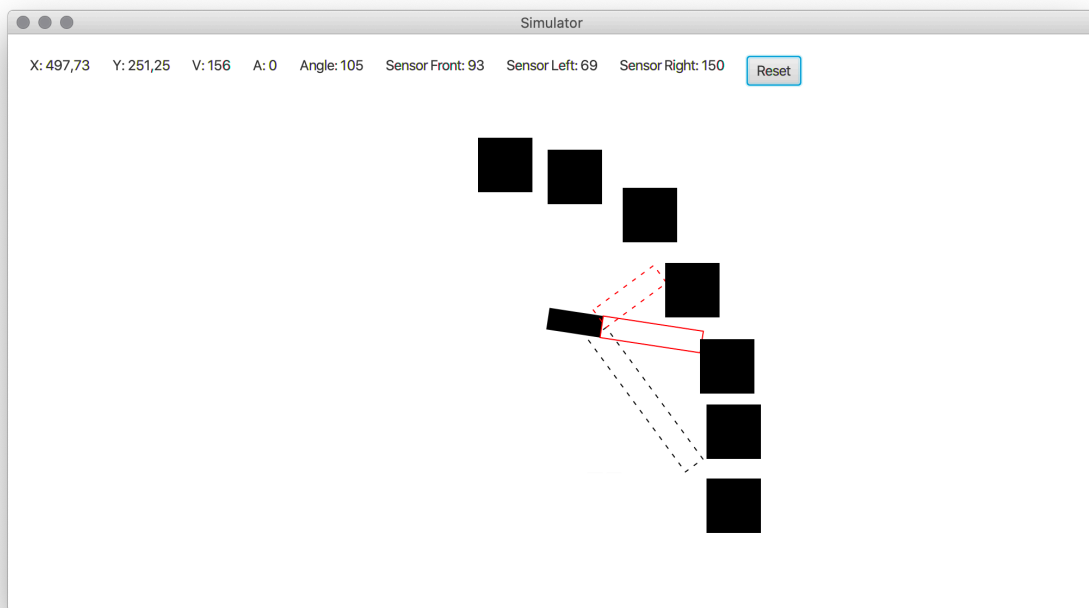


Abbildung 5: CrazyCar Simulator World

Das CrazyCar (siehe Abbildung 6) verfügt über drei Sensoren in der dargestellten Anordnung. Wird ein Hindernis vom Sensor erkannt, färbt dieser sich Rot und verkürzt sich entsprechend. Hindernisse können durch Klicken in die Welt hinzugefügt werden.

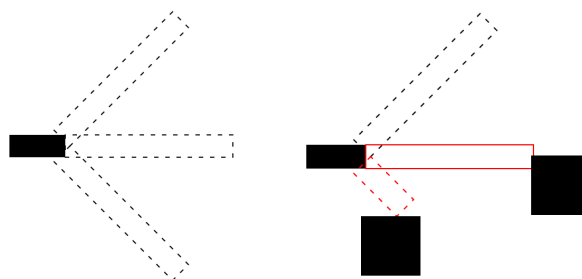


Abbildung 6: Virtual CrazyCar - ohne und mit Hindernis

7 DriveAssistan Funktionen

7.1 Nothalt

7.2 Ausweichen bitte!