Projektarbeit

Fahrsimulator

Sandro Ropelato (ropelsan)
Christof Würmli (wurmlchr)

12. Dezember 2011

Studiengang: Systeminformatik SI
Betreuender Dozent: Peter Früh (frup)

Inhaltsverzeichnis

1. Einleitung

1.1. Ausgangslage

Im Gebiet der Fahrsimulatoren gibt es bereits eine Vielzahl von verschiedenen Lösungen. Einige davon bestehen aus Filmmaterial, das abgespielt wird und der Fahrer muss auf die Bremse drücken sobald ein bestimmtes Ereigniss eintritt. Andere Fahrsimulationen bringen bereits eine virtuelle Welt mit, in der man sich mehr oder weniger frei bewegen bzw. frei fahren kann. Jedoch sind bei den meisten von diesen Fahrsimulatoren bereits feste Szenarien implementiert die nicht geändert werden können.

Die Grenzen eines Fahrsimulators liegen vor allem in der Leistungsfähigkeit des Rechners auf dem die Simulation installiert werden soll.

Das Projekt wird in Zusammenarbeit mit der ETH Zürich durchgeführt. Es ist bereits eine LabView Schnittstelle für das Steuerrad vorhanden.

1.2. Aufgabenstellung

1.2.1. Formulierung

Das Ziel der Arbeit besteht darin, einen Fahrsimulator für die bestehende Simulatinsumgebung zu erstellen. Diese besteht aus einem Fahrercockpit, einer Leinwand, einem Projektor und einem Computerterminal, von dem aus die Simulation gesteuert werden kann. Das Fahrercockpit enthält ein Steuerrad, drei Pedalen, einen Schaltknüppel und einen Autositz mit Sicherheitsgurt.

Die Fahrsimulation sollte dem Benutzer die Illusion des Autofahren möglichts realistisch vermitteln. Die soll durch einen Einsatz von Karteninfomationen von Google Maps oder Google Street View unterstützt werden.

Zudem sollen alle Betriebszustände und Benutzereingaben registriert und aufgezeichnet werden um eine genaue Analyse zu ermöglichen.

1.2.2. Aufteilung der Arbeit

Wir teilten die Arbeit im Wesentlichen in zwei Teile auf. Im ersten Teil legten wir den Fokus auf die korrekte Ansteuerung des Cockpits. Um dies zu testen, setzten wir uns das Ziel, ein Video abzuspielen und mit Gas- und Bremspedal die Geschwindigkeit kontrollieren zu können. Im zweiten Teil folgte die Implementation und Installation der Fahrsimulation.

1.3. Zeitplan und Arbeitsteilung

2. Aufbau des Systems

2.1. Systembeschreibung

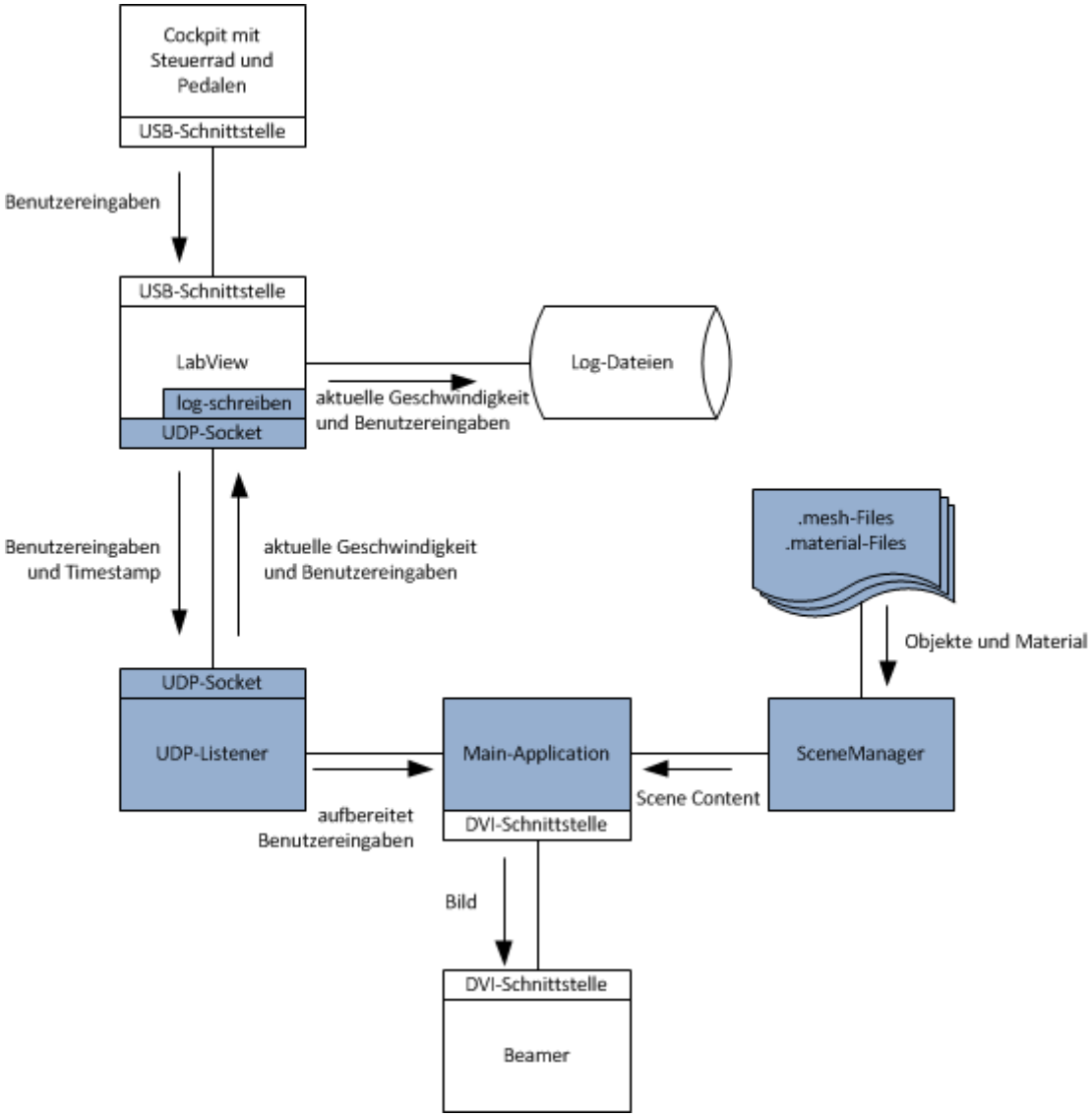

Abbildung 1: Systembeschreibung

Der Aufbau des Systems für den Fahrsimulator wird anhand der Abbildung 1 ilustriert. Die blau markierten Komponenten werden im Rahmen dieser Projekt Arbeit entwickelt. Alle übrigen sind bereits vorbestehend. Benutzereingaben, die im Cockpit gemacht wer-

den, werden von einem LabView Programm eingelesen. Nun benötigt es einen UDP-Port, über den verschiedenen Eingaben an das Programm weitergeleitet werden. Es handelt sich hierbei um Werte, die das Drehen des Steuerrades und den Druck auf Gas- oder Bremspedal quantifizieren. Zusätzlich wird der UDP-Port auch für das Empfangen diverser Log-Daten, die von unserem Programm gesendet werden, verwendet. Damit die Empfangenen Daten sauber in ein Log-File geschrieben werden, wird das LabVIEW Programm erweitert. Weiter muss in Programmiersprache C einen UDP-Socket mit entsprechendem UDP-Listener implementieren werden, um die Benutzereingaben zu empfangen. Gleichzeitig wird der UDP-Listener dazu verwendet die Geschwindigkeit des Fahrzeugs sowie Timestamps und weitere Daten an das LabVIEW Programm zurück zu schicken. Damit können die Daten gespeichert und später ausgewertet werden.
Diese Aufteilung durch eine Netzwerkschnittstelle ermöglicht es, das System, wenn notwendig, zu dezentralisieren. Einfachheitshalber wurde der UDP-Listener erst in einem Video-Beispiel implementier und getestet (Siehe Anhang B). Nachfolgend wird dieser UDP-Listener auch in das Programm des Fahrsimulator transferiert. Sind die Daten vom UDP-Listener empfangen und aufbereitet, werden sie im Hauptprogramm (Main Application) weiter verwendet. Während die Position des Steuerrades, des Gas und Bremspedales vom UDP-Listener permanent an das Hauptprogramm übertragen werden, wertet dieses die Positionen aus und veranlass die entspechenden Aktionen in der geladenen Szene. Die Szene selbst wird von einem Szenen-Manager geladen. Dieser benötig für die zu ladenden Objekte ein Meshfiles und mindestens ein Materialfile. Die Form jedes Objektes in der Szene wird durch ein separates Meshfile definiert. Texturen und Materialien werden durch ein oder mehrere Materialfiles beschrieben. Ein Materialfile kann zur Beschreibung unterschiedlicher Objekte verwendet werden. Die berechnete Szene wird schlussendlich in einem Fenster von Hauptprogramm angezeigt und über eine DVI-Schnittstelle an den Beamer übertragen. Der Beamer projeziert das Bild an die Wand, die sich direkt vor dem Cockpit befindet.

2.2. Anforderungen

2.2.1. Funktionale Anforderungen

- Der Proband kann das Fahrzeug im Fahrsimulator durch manipulation am Steuerrad und der Pedalen im Cockpit steuern.
- Die aktuelle Geschwindigkeit der Fahzeuges wird dem Probanden angezeigt.
- Es sollen zwei unterschiedliche Szenen zur verfügung gestellt werden. Die eine Szene sollte eine Stadt darstellen und die andere Szene eine Landschaft mit Tunnels.
- Die manipulationen des Benutzers und wichtige Parameter wie z.B. Geschwindigkeit sollen in einer Datei aufgezeichnet werden um diese auszuwerten.
- Alle ein- und ausgehenden Parameter des System sollen in LabVIEW verfügbar sein um diese auswerten und kontrollieren zu können.

2.2.2. Nicht Funktionale Anforderungen

a) Zuverlässigkeit
Das System soll robust sein.

b) Benutzbarkeit
Das Starten des Fahrsimulator sollte möglichst einfach gehalten werden. Das Steuer des Fahrzeuges soll möglichst intuitiev sein wie man es sich von einem richtigen Fahrzeug gewöhnt ist.

c) Aussehen und Handhabung
Der Fahrsimulator soll dem Probanden eine möglichst realistische Fahrsimulation bieten in der sich Strassen und verschiedene Objekte befinden.

d) Zeitliche Anforderungen
Die Raktionszeit des Systems soll möglichst klein sein. Die Verzögerung des Systems soll mess- und kalkulierbar sein.

e) Erweiterbarkeit
Das System soll möglichst Modular aufgebaut sein um später einfach erweitert werden zu können.

f) Portierbarkeit
Das System soll auf der existierender Hardware funktionieren, soll aber auch noch lauffähig sein wenn Teile des Fahrsimulators ausgetauscht werden.

Abbildungsverzeichnis

Anhang

A. Aufgabenstellung

B. Video Player

B.1. Ziel

Um manipulationen im Cockpit zu in unserem Programm zu empfangen zu zu verarbeiten, wird in einem ersten Schritt ein Video-Player realisiert der auf Eingaben des Probanden reagiert. Dies gibt ebenfalls einen ersten Eindruck von einem Fahrverhalten auf dem Simulator. Bei einem Video ist nur die Geschwindigkeit und noch keine Richtung und weiters relevant. Somit wird mit dem Video-Player die Ausgabe sehr reduziert. Ziel im Video-Player ist eine korrekte Übertragung der Eingaben im Cockpit über LabVIEW in das Programm. Das Programm soll die Parameter richtig interpretieren und entsprechende Aktionen veranlassen. Ein weiteres Ziel ist ein korrekte Rückgabe verschiedener Parameter an LabVIEW die von diesem in eine LogDatei geschrieben werden.

B.2. Systembeschreibung

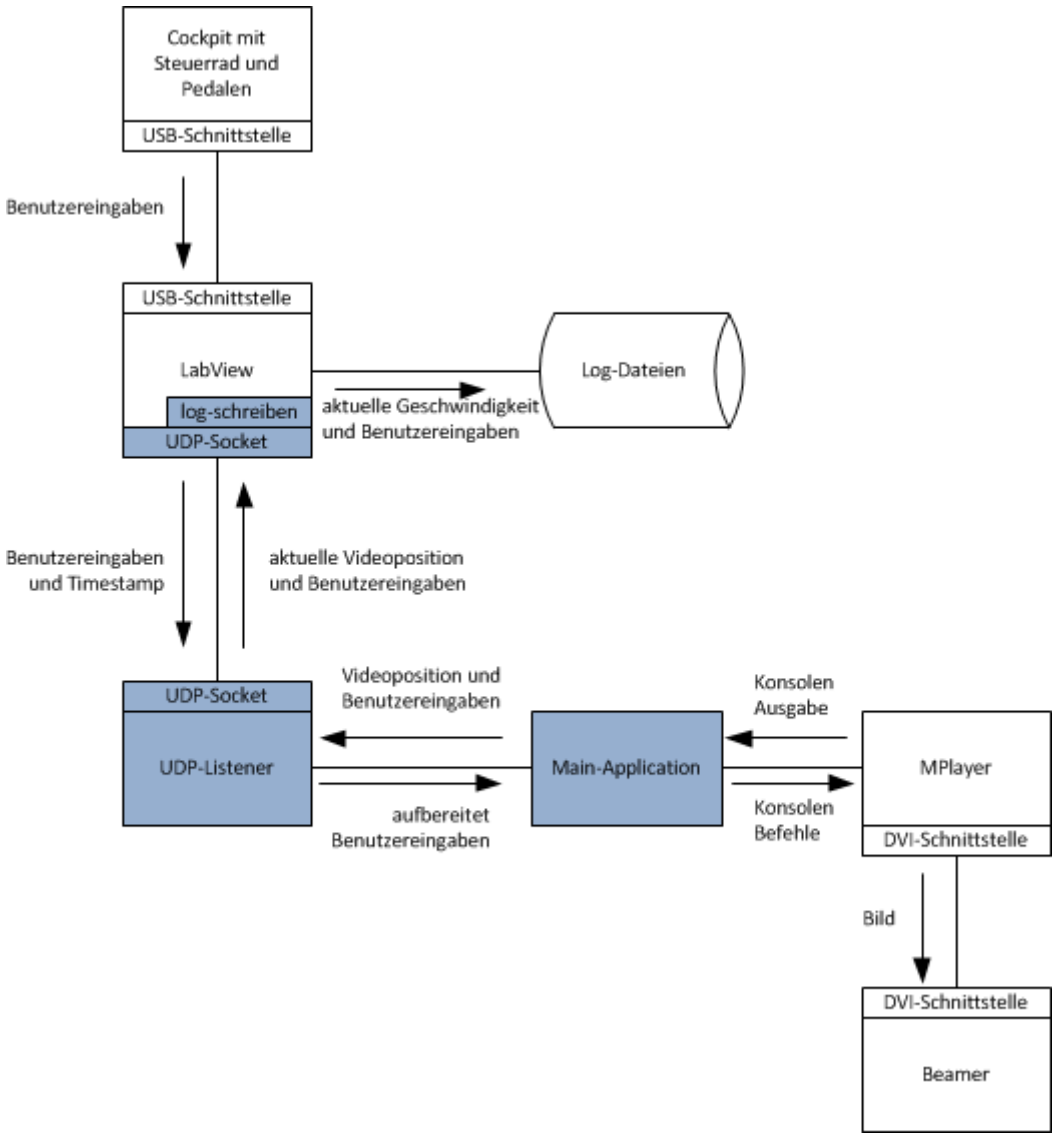

Abbildung 2: Systembeschreibung Video-Player

Um einzelne Komponenten des Video-Player wiederverwenden zu können, wird das System möglichst gleich wie das des Fahrsimulators aufgebaut. Dieser Aufbau wird anhand der Abbildung 2 illustriert. Die blau markierten Komponenten werden im Rahmen des Video-Players entwickelt. Alle überigen sind bereits vorbestehend oder werden installiert und verwendet. Die LabVIEW-Komponente in der Abbildung 2 liest bereits die Eingaben die der Proband im Cockpit mach ein. Dieses muss nun so erweitert werden, dass diese über einen UDP-Socket an das Programm übertragen werden. Zusätzlich muss im

LabVIEW ein UDP-Port eingerichtet werden, der verwendet werden kann, um UDP-Packete zu empfangen. Dieser wird benötigt um Daten die von unserem Programm gesendet werden in ein Log-File zu schreiben. Die Parameter werden vom vom UDP-Listener gespeichert und vom Hauptprogramm abgefragt. Aufgrund der Parameter wird dann die Geschwindigkeit des Videos manipuliert. Für das Abspielen des Videos wird der mPlayer verwendet. Die Befehle für den mPlayer können von unserem Programm über die Komandozeile abgesetzt werden. Ausgaben vom mPlayer werden ebenfalls über die Komandozeile den Standartausgang der Konsole übergeben.

B.3. Realisierung

In einem bestehendem LabVIEW-Programm werden bereits alle Eingaben, die im Cockpit gemacht werden können, eingelesen. Nun muss dieses Programm nur noch mit einem UDP-Port erweitert werden, damit es die Parameter, die unser Video-Player benötigt, senden kann. Diese Daten sind im Video-Player vor allem die betätigung von Gas- und Bremspedal. Diese beiden Pedalen werden in einem Koordinatensystem auf der y-Achse abgebildet. Die positive y-Richtung verifiziert das Gas und die negative y-Richtung das Bremspedal. Die intensität beider Pedalen wird durch 32767 bzw. 32768 ganzzahlige Werte identifiziert.

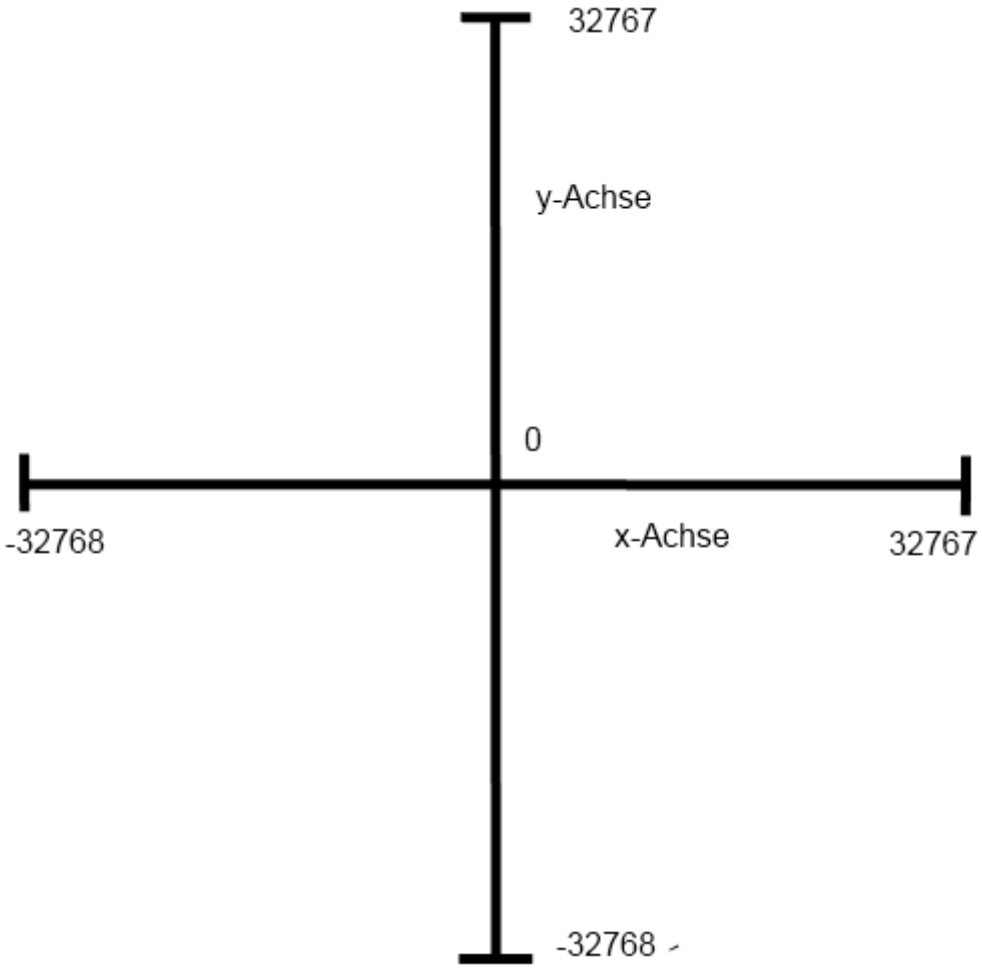

Abbildung 3: Koordinatensystem

Daten x und y verifiziert

C. Das Ogre Framework

D. Modelling Tools

E. Setup

F. Listings

G. Detailierter Zeitplan

H. Glossar

Lab-View Lab-View Programm Log-Datei Cockpit