

Projektarbeit

Fahrsimulator

Sandro Ropelato (ropelsan)
Christof Würmli (wurmlchr)

10. Dezember 2011

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

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 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 Simulationsumgebung 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 möglichs realistisch vermitteln. Die soll durch einen Einsatz von Karteninformationen 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

Der Aufbau des Systems für den Fahrsimulator haben wir uns wie in Abbildung 1 überlegt. Die Komponenten die blau markiert sind, müssen von uns entwickelt werden,

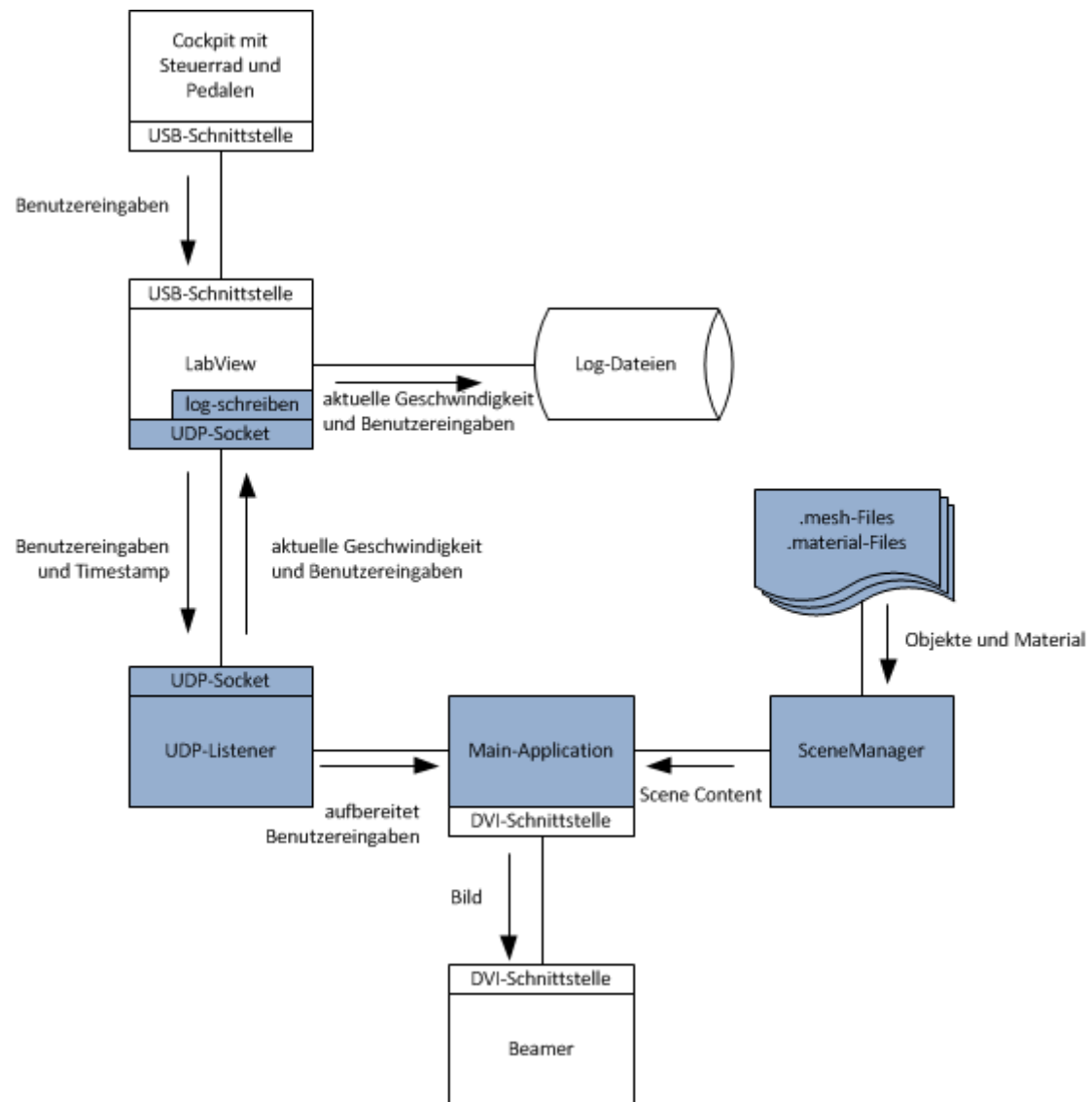


Abbildung 1: Systembeschreibung

die anderen bestehen bereits. Die Benutzereingaben die im Cockpit gemacht werden, werden bereits von einem LabView Programm eingelesen. Es benötigt lediglich noch einen UDP-Port über den wir die verschiedenen Eingaben an unser Programm weiterleiten können. Zusätzlich wird der UDP-Port auch für das empfangen verschiedener Log-Daten, die von unserem Programm gesendet werden, verwendet. Zu diesem Zweck muss das Lab-View Programm ebenfalls erweitert werden, damit die Empfangenen Daten sauber in ein Log-File geschrieben werden. Nun muss in C einen UDP-Socket mit entsprechendem UDP-Listener implementieren werden um die Benutzereingaben zu erhalten. Der UDP-Listener wird gleichzeitig verwendet um Geschwindigkeit des Fahrzeugs sowie Timestamp und weitere Daten an das Lab-View zurück zu schicken, damit diese gespeichert und später Ausgewertet werden können. Diese Aufteilung durch eine Netzwerkschnittstelle, ermöglicht es uns unser System, wenn notwendig, zu dezentralisieren. Dies wird im Realisierungskapitel des UDP-Listeners genauer ausgeführt. In einem ersten Schritt wurde der UDP-Listener in einem Video-Beispiel, nachzulesen im Anhang A, implementiert und getestet. Die Position des Steuerrades und der Gas und Brems-Pedalen werden vom UDP-Listener permanent das Hauptprogramm übergeben. Dieses wertet die Positionen aus und veranlasst die entsprechenden Aktionen in der geladenen Scene. Die Scene selbst wird von einem Scenen-Manager geladen. Dieser benötigt für die zu ladenden Objekte die Definition der Objekte in der Form eines mesh-Files und mindestens ein Materialfile in dem die benötigten Texturen und Materialien definiert sind. Die berechnete Scene wird in einem Fenster von Hauptprogramm angezeigt und über eine DVI-Schnittstelle an den Beamer übertragen. Der Beamer projiziert das Bild an die Wand die sich direkt vor dem Cockpit befindet.

2.2. Anforderungen

2.2.1. Funktionelle Anforderungen

2.2.2. Nicht Funktionale Anforderungen

- a) Zuverlässigkeit
- b) Benutzbarkeit
- c) Aussehen und Handhabung
- d) Wartbarkeit
- e) Zeitliche Anforderungen

3. Vorgehen und Methoden

3.1. UDP-Socket

3.2. Virtual Reality

3.3. Hauptprogramm

4. Resultate und Tests

4.1. Erreichtes

4.2. Zeitliches Verhalten des Systems

4.3. Testfälle

5. Nächste Schritte

5.1. Offene Punkte

5.2. Zusätzliche Funktionen

5.3. Ausblick auf Bachelor Arbeit 2012

6. Nachwort

6.1. Danksagung

7. Verzeichnisse

8. Anhang

A. Aufgabenstellung

B. Video Player

C. Das Ogre Framework

D. Modelling Tools

E. Setup

F. Listings

G. Detaillierter Zeitplan

Lab-View Lab-View Programm Log-Datei Cockpit