

Technische Dokumentation
zum Projekt



Ein IT-Systeme/Mobile Systeme-Projekt SS16

Teilnehmer:

Timo Böök (2103797)

Nicolas Graf (2094723)

Andreas Thomeßen (2102493)

Idee und Konzept

Treadmill Trophy gibt dem Spieler die Möglichkeit ein reelles RC-Modellauto per Smartphone auf einem Laufband zu steuern und mit virtuellen Gegenständen zu interagieren. Nach dem Highscore Prinzip gilt es den höchsten Punktestand zu erfahren indem man möglichst lange, unfallfrei weiterfährt. Die Spielfläche, die aus Fahrbahn und Hindernissen besteht, wird mit einem Beamer über einen Spiegel auf das Laufband projiziert. Ein Raspberry Pi 3 Model B (RPi) hostet die Serverumgebung um alles zu verbinden.

Wichtigste Elemente die also erstellt werden mussten sind:

- Laufband
- Modellauto
- Mechanische Steuerung des Modellautos
- Fernbedienungsoberfläche auf dem Smartphone
- Die zu projizierende Spielfläche
- Serverumgebung zum Interagieren von Fernbedienung, Spielfläche und Laufband
- RPi Konfiguration

Laufband

Das Laufband besteht aus zwei PVC-Rohren die im Abstand von ca. 70cm auf einer Holzplatte montiert sind, einem Plakat das über die zwei Rohre gespannt ist, und einem Motor der über einen Keilriemen das Laufband antreibt. Der Motor ist ein Hochleistungsgetriebemotor der über ein externes Netzteil mit 6.0V Betriebsspannung versorgt wird.

Ein zweiter Motor gleicher Art bewegt einen Keilriemen auf dem auf der oberen und unteren Seite jeweils ein Magnet angebracht ist. Der obere Magnet dient zum Ziehen des Autos von links nach rechts über einen Magneten zwischen den Vorderrädern. Der untere Magnet befindet sich dicht über einer Holzleiste auf dem drei Magnetfeldsensoren befestigt sind.

Sowohl die Magnetfeldsensoren als auch der zweite Motor sind mit dem RPi verbunden, somit lässt sich die Position des Autos bestimmen und verändern.

Querschnitt des Laufbands:

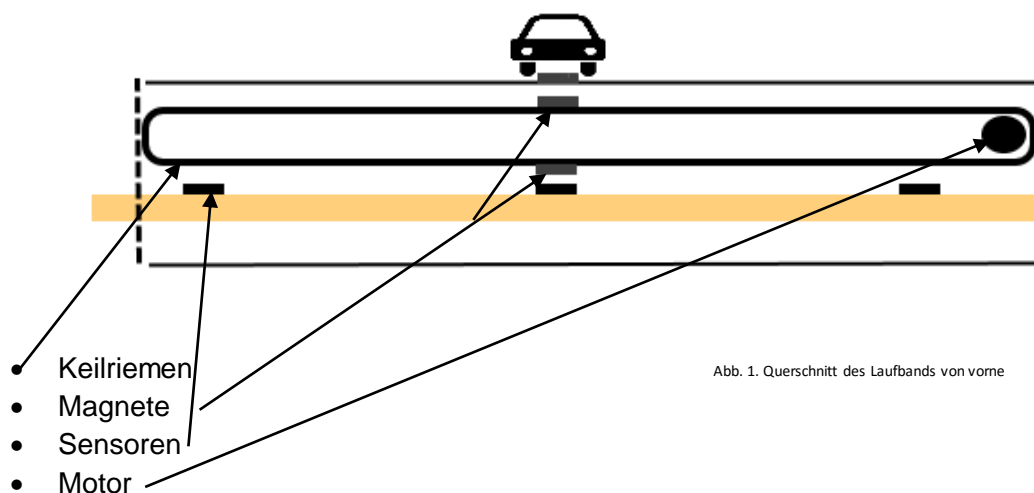


Abb. 1. Querschnitt des Laufbands von vorne

Fernbedienung

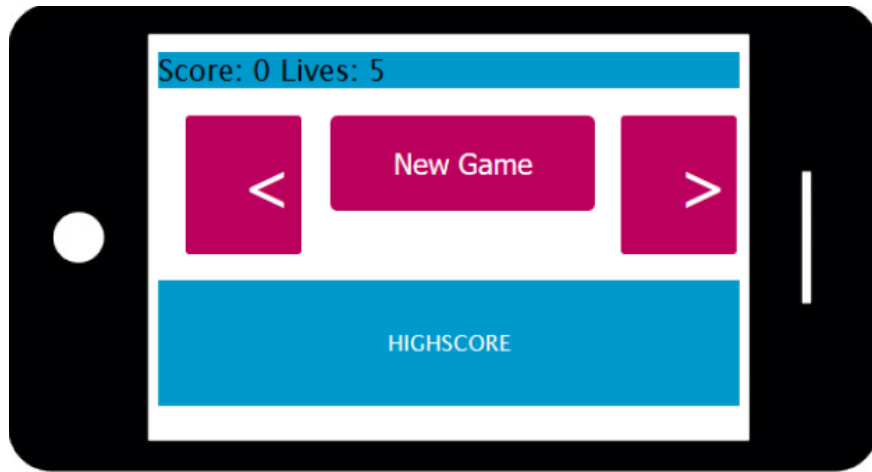


Abb. 2 Fernbedienungsoberfläche

Die Oberfläche die dem Spieler zur Verfügung gestellt wird besteht aus einer Punkteanzeige, einer Lebensanzeige und den Steuerelementen *Links*, *Rechts*, und *Neues Spiel*. Für den momentanen Highscore war das Layout schon erstellt, konnte aber aus zeitlichen Gründen nicht mehr implementiert werden.

Die hier gezeigte Ansicht erhält der Spieler, wenn er die IP Adresse des RPi's, auf dem der Node.js Server läuft, in seinen Browser eingibt. Das Smartphone verbindet sich also als Client und der Server stellt die HTML und CSS-Dateien zur Verfügung.

Beim Drücken des *Links*-Buttons schickt der Fernbedienungs-Client ein Event an den Server. Der Server liest nun die Werte der drei Magnetfeldsensoren aus um die Position des Autos herauszufinden. Wenn sich das Auto auf der mittleren oder rechten Spur befindet (also der mittlere oder linke Sensor ein Signal gibt), wird der Motor so lange nach links gedreht bis das Auto auf der linken Spur ist (rechter Sensor schlägt aus).

Gleiches passiert natürlich in angepasster Weise beim Drücken des *Rechts*-Buttons.

Auf Empfang des Events von dem *Neues Spiel*-Button wird die Spielfläche erstellt und das Spiel geht los.

Spielfläche

Zu Beginn des Projekts wurde eine erste Java Anwendung erstellt die die entscheidenden Funktionen beinhaltet und die grundlegende Logik darstellt.

Ablauf ist folgender:

Hindernisse entstehen am oberen Rand in einem bestimmten Zeitintervall. Der Spieler sitzt am unteren Bildrand und kann sich auf drei Spuren nach links und rechts bewegen. Erreicht das Hindernis den unteren Rand, verschwindet es und der Spieler erhält Punkte. Wenn er mit einem Hindernis kollidiert, verliert er ein Leben und das Hindernis verschwindet. Wenn die Lebensanzahl des Spielers auf null fällt ist das Spiel vorbei.

Gleiches galt es zu projizieren und die Position des Modellautos einzubeziehen. Dazu verbinden wir einen Computer (oder den RPi selbst) mit einem Beamer und greifen über den Server (vom RPi einfach localhost) auf den Ordner /game zu der uns die Spielfläche auf einer HTML Seite wiedergibt. Die Spiellogik ist also in JavaScript geschrieben und bearbeitet ein *canvas*-Element auf der HTML-Seite.

Die einfarbigen Quadrate tauschten wir im Laufe noch gegen selbst erstellte Grafiken aus. Der Untergrund ist mit Motion-Blur bearbeiteter Asphalt, und die Autos die uns entgegen kommen sind unser Modellauto abfotografiert mit veränderten Farben.

Alle Grafiken und Animationen die wir benutzen sind selbst erstellt.

Abb. 3. Runner.jar

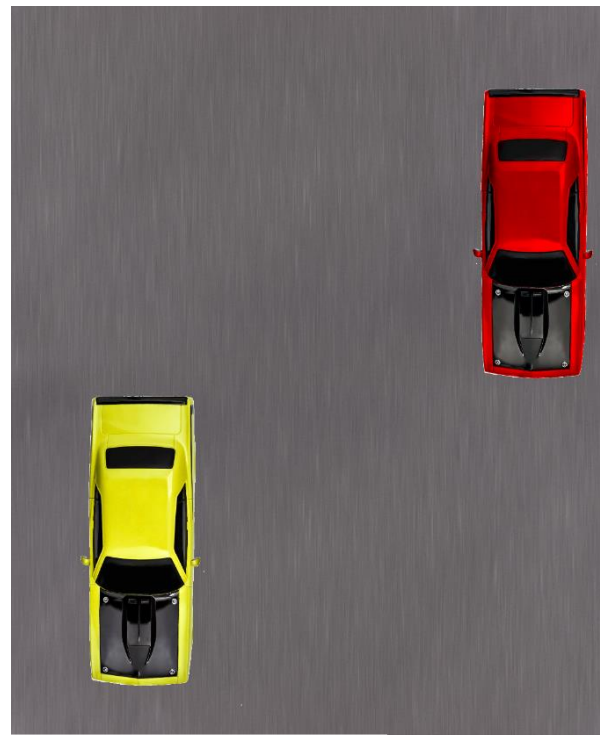
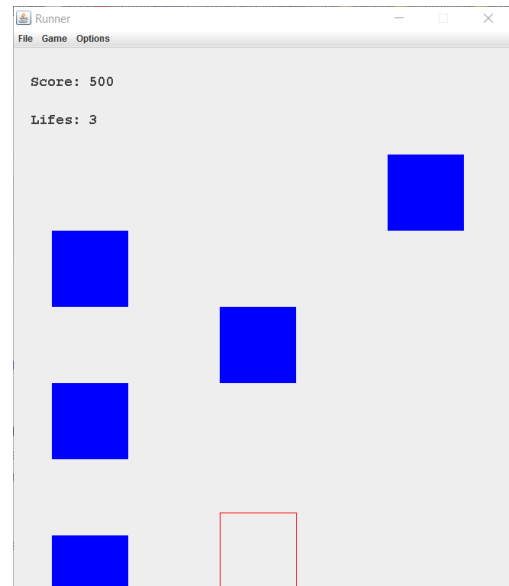


Abb 4. Fertige Spielfläche

Serverumgebung

Mithilfe der Plattform Node.js erstellen wir einen Webserver *TreadmillServer.js*. Dieser hört einen bestimmten Port ab und verbindet den Client mit Hilfe der „Socket.IO“ Library. Der Server selbst steuert auch den Motor der mit den GPIO-Pins des RPi's verbunden ist. Wir können mit der „onoff“ Library per JavaScript die Pins direkt ablesen und die Werte verändern:

```
//Drehen des Motors nach links
var Gpio = require('onoff').Gpio;

motor_input1 = new Gpio(4, 'out');
motor_input2 = new Gpio(17, 'out');
motor_enabler = new Gpio(18, 'out');

motor_input1.writeSync(1);
motor_input2.writeSync(0);
motor_enabler.writeSync(1);
```

Abb 5. Code Beispiel zur Motorsteuerung

Die GPIO-Pins die hier beschrieben werden sind mit einem Motortreiber-Chip (Modell: L293D) verbunden der wiederum direkt an die + und – Kontakte des Motors angeschlossen ist. (s. Schaltplan)

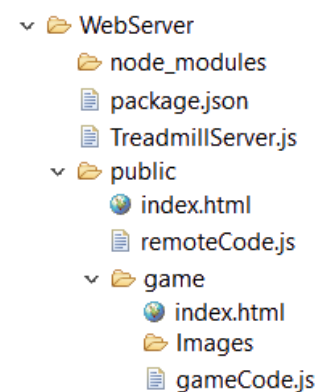


Abb. 6. Komplette Ordnerstruktur

Ausstellung

Am Ausstellungstag selbst wurde noch die Holzkonstruktion, die den Beamer und den Spiegel im richtigen Abstand voneinander fixiert, gebaut und ein lokales WLAN eingerichtet in dem sich der RPi und ein von uns gestellter Laptop befindet. Der Laptop ist an den Beamer angeschlossen und ruft vom RPi die Spielfläche ab. Der RPi selbst ist von der Performance leider zu schwach um unsere fertige Spielfläche flüssig wiederzugeben. Ein externes Netzteil versorgt den Motor der das Laufband antreibt direkt, und parallel unser Breadboard da der RPi selbst den Motor nicht antreiben kann. Als Fernbedienung benutzen wir entweder ein Smartphone von uns, oder alternativ kann sich ein Spieler mit seinem eigenen Smartphone erst mit dem WLAN und dann mit dem Server verbinden.

Probleme + Fazit

Von dem Feintuning und der Erhöhung der Belastbarkeit des Laufbands abgesehen boten sich große Schwierigkeiten bei der Verkabelung. Bei der eindeutig labilsten Stelle der Konstruktion, nämlich auf dem Breadboard, konnte es passieren, dass ein Signal nicht übertragen wird weil ein Kabel nicht ganz am Kontakt sitzt. Herauszufinden ob es wirklich daran liegt und zusätzlich noch welches Kabel genau, gestaltete sich jedes Mal als schwierig. Zusätzlich hielt der eine Motor den Belastbarkeitstest des Ausstellungstages nur bedingt stand, und brauchte mehrere Kühlungspausen.

Alles in allem fanden wir die Zusammenarbeit von MT und MS sinnvoll, es hat uns allen dreien großen Spaß gemacht, und wir hoffen, dass dieses Format für spätere Semester beibehalten wird.