

Semesterarbeit im Fach User Interfaces

Entwicklung eines User Interfaces für einen serverbasierten, 3D
gedruckten Roboterarm

Eingereicht am: 24. Juli 2020

Josef Prothmann
geboren am 16. Dezember 1998
Email: j.prothmann@stud.hs-wismar.de

David Oliver Taube
geboren am 9. Februar 1994
Email: d.taube@stud.hs-wismar.de

Betreuer: Prof. Dr. H. Litschke

Abstrakt

Diese Arbeit beschäftigt sich mit der Implementierung eines User Interfaces für einen Roboterarm, welcher von einem 3D Drucker hergestellt und über einen Server mit einer Android Applikation zur Fernsteuerung verbunden ist.

Abstract

This thesis is facing the issue of implementation an user interface for an robotic arm wich is crafted by a 3D printer and is connected to a server while it is remote controlled by an android application.

Inhaltsverzeichnis

1	Einleitung	2
2	Grundlagen	3
2.1	Hardware Komponenten	3
2.2	3D - Druck	4
2.2.1	Beispieldruck PLA	5
2.3	Android	5
3	Konzeption	6
4	Implementierung	7
4.1	Client	7
4.2	Roboterarm	8
4.3	Server	9
5	Fehlschläge	11
6	Zusammenfassung	i
	Literaturverzeichnis	ii
	Abbildungsverzeichnis	iii
	Selbstständigkeitserklärung	iv

1 Einleitung

Die Benutzerschnittstelle (engl. User Interface (UI)) gewinnt zunehmend an Aufmerksamkeit. Dies resultiert aus der laufenden Digitalisierung von Prozessen und der stetigen Übernahme der Computer und Maschinen von menschlichen Aufgaben. Mittlerweile sind UIs nicht mehr wegzudenken, denn sie treten in Form von Applikationen (App) auf dem Smartphone, als Terminal auf dem PC oder als Spracheingabe im Auto auf. In diesem Projekt wurde eine Benutzerschnittstelle für einen Roboterarm als Android App entwickelt, mit dem sich dieser steuern lässt.

2 Grundlagen

2.1 Hardware Komponenten

Die Technische Hardware in diesem Projekt besteht aus 3 Komponenten. Dem Raspberry PI, den Servomotoren und dem Servo-Treiber, auf diese wird im Folgendem genauer eingegangen.

Raspberry PI

Der Raspberry PI ist ein Einplatinencomputer, mit dem dafür zugeschnittenem Betriebssystem „Raspbian“, einem Debian Derivat. Dieser bietet trotz des günstigen Preises ein herkömmliches, voll funktionsfähiges Computersystem bestehend aus CPU, Arbeitsspeicher, Sekundärspeicher und diversen Schnittstellen wie HDMI, AUX und USB. Mit Hilfe von Ansteuerungspins ist es möglich, externe Komponenten anzuschließen, wie zum Beispiel Sensoren, mit denen es möglich ist, Informationen zu empfangen. Dadurch sind diverse Projekte, wie zum Beispiel eine Aquariumsanzeige, SmartHome Steuerungen oder simple „Roboter“, möglich.



Abbildung 2.1: Raspberry Pi Modell 3b - *Quelle: raspberrypi.org [1]*

Servomotoren

Servomotoren sind spezielle Elektromotoren, welche die Möglichkeit besitzen, die Winkelposition ihrer Motorwelle sowie die Drehgeschwindigkeit und Beschleunigung zu kontrollieren. Sie bestehen aus einem Elektromotor, der zusätzlich mit einem Sensor zur Positionsbestimmung ausgestattet ist. Die durch den Sensor ermittelte Drehposition der Motorwelle wird kontinuierlich an eine, meist außerhalb des eigentlichen Motors angebrachte, Regelelektronik übermittelt, den so genannten Servoregler, welcher die Bewegung des Motors entsprechend einem oder mehreren einstellbaren Sollwerten, als auch Soll-Winkelposition der Welle oder Solldrehzahl in einem Regelkreis regelt.

16-channel 12-bit Servo Driver

Dieses Modul bietet eine Steuerung von mehreren Servo Motoren gleichzeitig. Durch eine externe Spannungsquelle sind bis zu 16 5V Servo Motoren anschließbar.

2.2 3D - Druck

3D Druck ist eine Technologie die seit Jahren von der Industrie als auch von Privatendwendern betrieben, verbessert und erweitert wird. Diese Art des Druckens umfasst sämtliche Verfahren in denen Material Schicht auf Schicht aufeinander aufbauend gedruckt wird. Für dieses Verfahren wird ein 3D Drucker benötigt. Ein handelsüblicher Filament (thermoplastische Kunststoffe in Drahtform) Drucker besitzt in der Standardvariante drei bis vier Achsenbewegungsmotoren, einen Extrudermotor, ein Heizbett, ein Controllerboard, ein Hotend und einen Rahmen indem alle Bauteile verankert sind, sowie eine Spannungsquelle. Für das Drucken von 3D Objekten sind neben einem Drucker auch Druckmaterial sowie ein Slicingprogramm nötig. Je nach Druckerart können verschiedenste Filamente verwendet werden. Das am meistverwendete Material ist PLA(Polylactide). Diese Polymere zählen zu den Polyestern und lassen sich am einfachsten drucken.

In dem Slicingprogramm werden die Modelle auf einem virtuellem Druckbett positioniert und in einzelne Schichten zerlegt. Je nach dem wie hoch eine Schicht ist, desto detaillierter ist das gedruckte Produkt. Aktuelle Drucker sind in der Lage eine Schichthöhe von 0,1 mm zu erreichen. Das bedeutet, dass für ein 1 mm hohes Objekt 10 Schichten für eine Schichthöhe von 0,1 mm verwendet werden müssen und für eine Schichthöhe von 0,2 nur 5.

Der Druckkopf(Hotend), aus dem das heiße Filament kommt, bewegt sich im dreidimensionalen Raum(drei Achsen, X,Y,Z). Diese Bewegungen werden durch die Achs-

bewegungsmotoren ausgeführt. Um das Material in das Hotend(über 200 Grad heißer Metallblock mit Düsenspitze) zu drücken wird ein Extrudermotor benötigt. In dieser Beschreibung wurden Fused Deposition Modeling / Fused Filament Fabrication (FDM / FFF) besprochen. Daneben gibt es noch andere Druckerarten, wie zum Beispiel Stereolithographie (STL / SLA), bei welchem Flüssigharz und UV-Strahlen verwendet werden um Modelle zu drucken.

2.2.1 Beispieldruck PLA

Nachdem der Drucker die Datei mit dem gesliceten Druckmodel geladen hat, findet die Vorbereitung des Druckers automatisch statt. Das Druckbett bzw. Heizbett wird auf 60°C vorgeheizt, damit das Material besser an der Druckfläche haftet. Währenddessen wird das Hotend auf eine Betriebstemperatur von ca 215°C erhitzt, damit das Material formbar wird und Schicht für Schicht aufgetragen werden kann. Sind die Endtemperaturen erreicht, werden die einzelnen Befehle aus der gesliceten Datei geladen. Diese beinhaltet Befehle für die Motoren. Die gedruckte Schicht im Druckbett verhärtet sofort, sodass die nächst höhere Schicht auf eine stabile Oberfläche gesetzt werden kann.

2.3 Android

Android ist ein von Google Inc. entwickeltes und das meist verbreiteste Betriebssystem auf dem deutschen Markt¹. Mittlerweile findet es nicht mehr nur in Smartphones, sondern auch in Smartwatches, Tablets und in Fernsehern Verwendung. Die App wurde auf der Version 8.1 (Oreo) entwickelt, ist aber trotzdem zu aktuelleren Versionen kompatibel.

Die aktuellste Android Version ist Android 10. Android 11 folgt im dritten Quartal 2020.

¹Stand März 2020 laufen ca. 80% der verfügbaren Geräte mit einem Android Betriebssystem -Quelle: Statista.com, „Vergleich der Marktanteile von Android und iOS am Absatz von Smartphones in Deutschland von Januar 2012 bis März 2020“, 03.06.2020 [2]

3 Konzeption

Bei dem Entwurf des Graphical User Interfaces (GUI) wurde sich an gängigen Fernsteuerungen aus der Spielbranche und dem Modellbau orientiert. Auf der linken sowie auf der rechten, unteren Seite des Displays, lässt sich die Position des Arms im dreidimensionalen Raum (3D) steuern. Im oberen Bereich der Anwendung befinden sich Konfigurationspunkte sowie der „Switch“², welcher das Steuern des Greifers übernimmt. Die App sollte nur im „Landscape“-Modus, ausführbar sein, dies hat technische- und ergonomische Hintergründe. Dabei wurde Wert auf das Einhalten von aktuellen Konventionen, wie einem Settings Menü zum Festlegen von Serverport und -IP Adresse, als auch auf CISCO Standard Symbolen wie dem Netzwerk-Symbol geachtet. Des Weiteren wurde mit dem Erstellen von eigenen Scaleable Vector Graphics (SVG) die Intuitivität unterstützt und eine missverständliche Nutzung der Applikation ausgeschlossen.

Für die Kommunikation zwischen der App und dem „Roboterarm“ soll ein Server als Verbindungskomponente verwendet werden. Mit diesem Konzept wird eine asynchrone Kommunikation zwischen Server und Client ermöglicht. Der Client, welcher als Sender von Daten dient, kann dem Server, welcher die Daten empfängt, direkt Daten übermitteln. Diese Verbindung wird durch einen TCP Socket gewährleistet. Die gesendeten Informationen der Applikation werden über die auf dem Raspberry PI vorhandenen GPIO Pins an die Servomotoren mittels eines „16-channel 12-bit Servo Driver“ übertragen. Dieser Treiber ist Notwendig um eine konstante und ausreichende Spannungsquelle für die Motoren zu gewährleisten. Eine direkte Ansteuerung von mehr als einem Motor über die Spannungsquelle des Raspberry PI's ist ungenügend und lässt die Servomotoren zucken oder gar nicht reagieren.

²Ein Switch ist in Android ein Kippschalter mit 2 Zuständen

4 Implementierung

4.1 Client

Für die Entwicklung einer nativen Android App wird die Entwicklungsumgebung Android Studio benötigt. Des Weiteren wird für die Softwareversionsverwaltung auf die Plattform Github zurückgegriffen, wo der Quellcode des Projektes dokumentiert ist³. Aus dem Grund, dass die Verbindung zu dem Roboter Arm über das Internet realisiert wurde, ist es notwendig in dem Manifest der Anwendung die Internetberechtigung zu erteilen.

```
<uses-permission android:name="android.permission.INTERNET" />
```

Damit ist es möglich auf das mobile Netz des Smartphones zuzugreifen. Um eine Verbindung aufzubauen, benötigt der Client, in diesem Fall die App, die Adresse und den Port des Hosts, in diesem Fall der Server auf dem Raspberry Pi.

Die Parameter für die Verbindung (IP und Port des Hosts), lassen sich dynmaisich (zur Laufzeit) in der App einsehen und anpassen. Dadurch ist es in der Theorie möglich mehrere Roboter nacheinander zu steuern. Damit das ankommende Signal welches die Knöpfen und der Switch senden, korrekt interpretiert wird, sendet die App einen eindeutigen, einstelligen Integer Code. Dadurch lassen sich die Bewegungen der Motoren durch den korrekten Knopf in der App abbilden.

Da Pfeile als Bewegungsindikatoren im 3D Raum leicht missverstanden werden können, wurden eigene SVGs erstellt, welche die Bewegung des Arms verdeutlichen.

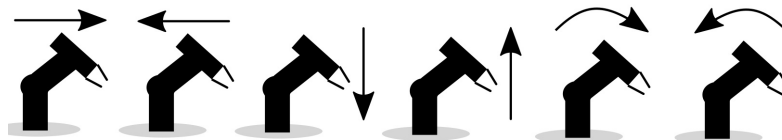


Abbildung 4.1: SVG der Knöpfe aus der App - *Quelle: Eigene Darstellung*

³<https://github.com/JoProt/User-Interface-Project>

SVG hat den Vorteile, dass die Kanten und Formen in der Bildebene mathematisch berechnet werden. So lässt sich diese Grafik beliebig skalieren.

In Abbildung 4.1 sind die einzelnen Grafiken für die jeweilige Bewegung des Roboters aufgelistet, wobei \leftarrow und \rightarrow für Steuerung auf einer horizontal Linie, \uparrow und \downarrow für vertikale Bewegungen und \curvearrowleft und \curvearrowright für Bewegung um die eigene Hochachse verantwortlich sind.

4.2 Roboterarm

Der Arm wurde von einem selbstgebauten 3D-Drucker über 25 Stunden gedruckt und hat durch die Ungenauigkeit des Druckers und falschen Einstellungen einen leichten Verzug in den Bauteilen. Das bedeutet, die Greifzange hat ein leichtes Problem beim Schließen, dies würde sich mit neuen gedruckten Greifzangen-Bauteilen beheben lassen. Aus technischen Gründen ist dies nicht mehr möglich gewesen, da der Drucker derzeit einem technischen Defekt unterliegt. Der Arm hat an vier Punkten die Servomotoren implementiert und kann sich dadurch wie in Abschnitt 4.1 beschrieben in mehrere Richtungen bewegen. Die Motorsteuerung funktioniert über ein 16-channel 12-bit Servo Driver-Board. Genaueres zu diesem Bauteil ist unter dem Abschnitt 2.1 zu lesen.

4.3 Server

Für den Server und die Ansteuerung des Roboterarms wurde die Programmiersprache Python verwendet [3]. Die Verbindung zum Server wird über eine Socket Verbindung gewährleistet, dass bedeutet, es wird eine direkte Verbindung zwischen Client und Server aufgebaut, die solange aktiv ist bis einer der beiden Partner die Verbindung trennt. Eine aktive Trennung findet in dieser Implementierung nur durch den Client statt. Wie im folgenden Codeauszug zu sehen ist wird eine Verbindung aufgebaut und gehalten. Es werden Daten empfangen, welche an die Roboterarmsteuerungsmethode weitergeleitet werden.

```
# Server Init and Loop ——
serv = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serv.bind(('192.168.178.29', 5556))
serv.listen(5)
print("Server started")

while True:
    conn, addr = serv.accept()
    print('client connected')
    from_client = ''
    while True:
        data = conn.recv(4096)
        if not data: break
        print(data.decode())
        handleData(data.decode())
    conn.close()
    print('client disconnected')
```

In der Armsteuerung wird überprüft, welche Daten gesendet wurden. Intern wurde sich zuerst über eine passende Kommunikation ausgetauscht und danach wurden die jeweiligen Bewegungen per Zahl definiert. Beispielsweise wird Basis, welche das drehen nach links und rechts ermöglicht durch Integerwerte 7(links) und 8(rechts) definiert. Im folgenden Codeausschnitt wurde vorher überprüft ob es sich wirklich um eine Zahl handelt, dann wurde sie geparkt und per „if“-Abfrage wird nun geprüft um welche Zahl es sich handelt.

```
if (number==8):
    if (servoPosBase+servoSteps<=maxServo):
        servoPosBase+=servoSteps;
        kit.servo[2].angle = servoPosBase;

if (number==7):
    if (servoPosBase-servoSteps>=minServo):
        servoPosBase-=servoSteps;
        kit.servo[2].angle = servoPosBase;
```

Hierbei ist zu erkennen, dass der Mindest- und der Maximalwinkel des Servomotors als auch die Schrittgröße für ein Eingangssignal vorher definiert wurden. Dies dient dazu Fehler für die Motoren zu verhindern. Mithilfe des “ServoKit”-Packages ist es möglich eine vorgefertigte Ansteuerungsmethode zu verwenden [1].

Wenn ein Datenstrom vom Client gesendet wird, wird dieser durch den Server validiert und weitergereicht um einen Motor des Armes zu bewegen.

5 Fehlschläge

Für die Server-Arm Steuerung wurden mehrere Ansätze ausgetestet. Zum Einen war das Wissen, wie solche Motoren und Systeme funktionieren nicht vorhanden. Dadurch sind Probleme wie zuckende, nicht reagierende Servomotoren und Fehlansteuerungen entstanden. Erst bei der Betrachtung, dass die Spannungsquelle ungenügend sein könnte, wurde recherchiert [4] und Lösungen betrachtet.

6 Zusammenfassung

In diesem Projekt haben wir uns mit den Grundlagen der Entwicklung eines User Interfaces auseinandergesetzt und eine ansprechende und intuitiv zu bedienende App entwickelt. Mit Hilfe unseres Wissens aus weiteren Modulen, wie zum Beispiel „Programmierung Mobiler Endgeräte“ und „Echtzeit- und Netzwerkprogrammierung“, war es uns möglich, eine Android Anwendung zu schreiben um den Roboter Arm zu steuern.

In der Zeit der Digitalisierung, in der Computer immer mehr Teil unseres Alltags werden ist es das Ziel, dass die Maschinen für den Menschen eine Arbeitserleichterung und keine zusätzliche Last werden. Deshalb ist es umso wichtiger, mit einem qualitativem User Interface zu arbeiten. Dieses vergleichsweise kleine Projekt hat uns gezeigt, wie aufwendig der Prozess der Gestaltung einer hochwertigen Oberfläche ist.

Gerade dann, wenn eine Anwendung veröffentlicht und von mehreren Menschen genutzt wird, ist es wichtig, dass es das Leben erleichtert und nicht verkompliziert.

Literaturverzeichnis

- [1] FOUNDATION, Raspberry P.: Raspberry Pi 3 Model B. In: *www.raspberrypi.org* (2020)
- [2] TENZER, F.: *Marktanteile von Android und iOS am Smartphone-Absatz in Deutschland bis März 2020.* <https://de.statista.com/statistik/daten/studie/256790/umfrage/marktanteile-von-android-und-ios-am-smartphone-absatz-in-deutschland/>. Version: Mai 2020
- [3] CHARLY KÜHNAST, Michael K. u.: *Raspberry Pi: Das umfassende Handbuch für Maker und Tekkies. Aktuell zu Raspberry Pi Model 3B+ und Zero WH.* Deutschland : Rheinwerk Computing, 2018
- [4] TOWNSEND, Kevin: *Configuring Your Pi for I2C.* <https://learn.adafruit.com/adafruit-16-channel-servo-driver-with-raspberry-pi/configuring-your-pi-for-i2c>. Version: Mai 2020

Abbildungsverzeichnis

2.1	Raspberry Pi Modell 3b - <i>Quelle: raspberrypi.org [1]</i>	3
4.1	SVG der Knöpfe aus der App - <i>Quelle: Eigene Darstellung</i>	7

Selbstständigkeitserklärung

Hiermit erklären wir, dass wir die hier vorliegende Arbeit selbstständig, ohne unerlaubte fremde Hilfe und nur unter Verwendung der aufgeführten Hilfsmittel angefertigt haben.

Wismar, den 24. Juli 2020

Ort, Datum

Unterschrift