

Inhaltsverzeichnis

1. EINFÜHRUNG

**1.1. Problemstellung und Umstände**

**1.2. Unsere Lösung für das Problem**

2. ART UND WEISE DES VORGEHENS

3. UMSETZUNG

**3.1. Das Entwickeln eines Roboters**

3.1.1. Prototyp 1

3.1.2. Prototyp 2

3.1.3. Prototyp 3

3.1.4. Prototyp 4

3.1.5. Prototyp 5

3.1.6. Prototyp 6

**3.2. Hardware und Software des EV3**

3.2.1. Die Hardware

3.2.2. Die Software

**3.3. Die Applikation**

3.3.1. Verbindung von Smartphone und EV3

3.3.2. Weitere Funktionen der Applikation

**3.4. Einbindung des Kinect v2 Sensors**

**3.5. Verbindung von Laptop und EV3**

4. ERGEBNISSE

5. DISKUSSION

**5.1. Probleme bei der Umsetzung**

**5.2. Weiterentwicklungsmöglichkeiten**

6. DANKSAGUNG

7. QUELLEN- UND LITERATURVERZEICHNIS

Abbildungsverzeichnis

Abbildung 1:

Abbildung 2:

Abbildung 3:

Abbildung 4:

Abbildung 5:

Abbildung 6:

Abbildung 7:

Abbildung 8:

Abbildung 9:

Abbildung 10:

1. EINFÜHRUNG

* 1. **Problemstellung und Umstände**

Der demographische Wandel unserer Gesellschaft ist gekennzeichnet von sinkenden Geburtszahlen, erhöhter Lebenserwartung und Auflösung von traditionellen Familienstrukturen. Ältere und körperlich eingeschränkte Menschen sind vermehrt auf sich alleine gestellt und auf Unterstützung angewiesen.

Dabei ist das Transportieren schwerer Gegenstände, wie zum Beispiel Einkäufe, oftmals ein Problem von großer Bedeutung. Aufgrund dieser Umstände können vor allem alleinlebende, körperlich eingeschränkte Menschen sich nicht mehr selbst versorgen und sind auf Unterstützung zum Beispiel durch Haushaltshilfen angewiesen, welche viel Geld kosten, das selten aufgebracht werden kann.

Auch moderne Online-Dienstleistungen in Form von Lieferservices entsprechen häufig nicht den Bedürfnissen älterer Menschen, da diese sich in den wenigsten Fällen ausreichend mit Computern auskennen oder keinen Internetzugang besitzen.

* 1. **Unsere Lösung für diese Problematik**

Um das Problem, welches das Tragen von schweren Einkäufen oder Ähnlichem darstellt, zu lösen entschieden wir uns einen Roboter zu konstruieren welcher diese Aufgabe übernehmen sollte. Dieser sollte, um für die meist technikunerfahrene ältere Generation leicht handhabbar zu sein, autonom und somit ohne weitere manuelle Steuerung, seinem Besitzer folgen und sonst nur durch Sprachbefehle gesteuert werden. Außerdem muss der Roboter sich den Gegebenheiten, wie zum Beispiel variablen Geschwindigkeiten unterschiedlicher Menschen anpassen können. Dazu kommen noch einige weitere Grundvoraussetzungen, die es zu beachten gibt, wie beispielsweise eine geeignete Höhe der Tragefläche, um ein anstrengendes Bücken des Besitzers zu vermeiden.

2. ART UND WEISE DES VORGEHENS

Um alle Erforderlichkeiten, die für einen guten Roboter nötig sind, zu erfüllen, planten wir eine ganze Reihe von verschiedenen Modellen, wobei die verwendete Technik und Software sich, auf Basis der vorherigen Modelle, immer weiter verbessern sollte.

Durch dieses System setzten die ersten Prototypen noch auf sehr einfache Technik sowie Mechanik und es fehlten einige der oben genannten Funktionen.

Dabei testeten wir verschiedene Arten der Fortbewegung und der Positionsbestimmung und nutzten verschiedene Programmiersprachen und Bibliotheken, bis wir uns für eine finale Methodik entschieden.

Mit der Zeit sollten außerdem noch einige weitere, zusätzliche Funktionen hinzukommen, um den Nutzern des Roboters den Alltag zusätzlich zu erleichtern.

Nach dem Bau eines Prototyps erstellten wir ein passendes Programm, immer an die Fähigkeiten des gegenwärtigen Roboters angepasst. Dabei teilten wir das gesamte Projekt in mehrere Themenbereiche ein, welche sich jeweils mit einer einzelnen Thematik befassen. Dies sorgte sowohl für einen besseren Überblick, als auch für die Möglichkeit sich genauer mit einzelnen Problemen zu beschäftigen. Eine solche Vorgehensweise braucht zwar viel Zeit, überzeugt jedoch mit durchdachteren Ergebnissen und einer klaren und effizienten Strukturierung.

3. UMSETZUNG

**3.1. Das Entwickeln eines Roboters**

Bei der Entwicklung der Prototypen gingen wir wie oben beschrieben vor (siehe 2.). Das bedeutet, dass wir insgesamt sechs Prototypen entwickelten, um uns einem optimalen Roboter schrittweise anzunähern. Beim Material setzten wir aufgrund der hohen Flexibilität und akzeptabler Stabilität auf die „LEGO-Technic“ Bauteile.

3.1.1. Prototyp 1:

Unser erstes Konstrukt war nicht viel mehr als ein simples Fahrgestell. Es konnte durch seine zwei gefederten Ketten, welche ohne Getriebe direkt an je einen Motor angebunden waren, nur sehr langsam und nur geradeaus fahren. Daran diesen zu programmieren war nicht einmal zu denken, da noch kein Computer verbaut war, welchen man hätte programmieren können.

3.1.2. Prototyp 2:

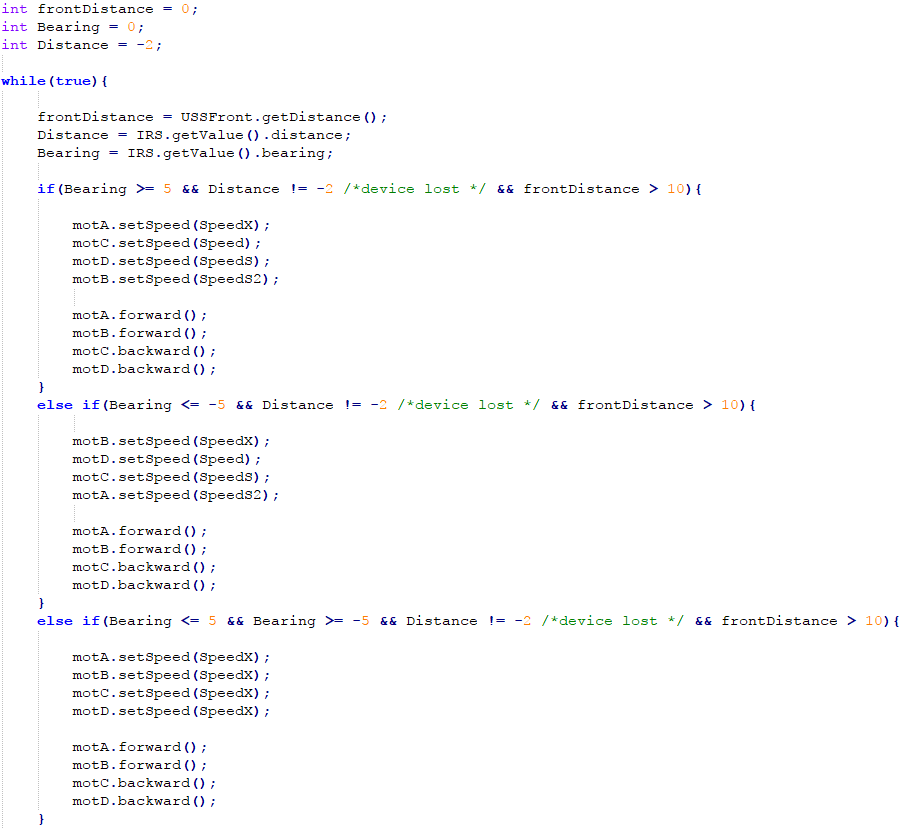
Bei unserem zweiten Modell tauschten wir die Ketten gegen Räder aus, bauten einen EV3 (siehe 3.2.1.) ein und fügten ein pneumatisches Grundsystem (elektronische Pumpe, Drucklufttank, einige Kolben) hinzu mit dem wir weitere Funktionen, wie eine verschließbare Klappe für den Einkaufskorb umsetzen wollten. Dieser Prototyp konnte sich dann, mithilfe eines kleinen, in der grafischen LEGO Umgebung erstellten Programm, grundlegend bewegen. Da allerdings, noch keine Sensoren verbaut waren, war er weder in der Lage Objekten auszuweichen, noch einer Person zu folgen.

3.1.3. Prototyp 3:

Bei dem dritten Prototyp unseres Projekts versuchten wir erstmals den Roboter Treppen steigen zu lassen. Wir setzten dabei auf ein System, bei dem sich der Roboter durch vier an den Seiten angebrachte Arme auf Höhe einer Treppenstufe stemmt. Dazu konnte dieser alle Arme nach oben und nach unten schwenken. Zum Fahren waren alle vier Arme nach unten geklappt, sodass die angebrachten Räder auf dem Boden auflagen. Das dafür entwickelte Programm sollte dafür sorgen, dass der Roboter zuerst an eine Stufe heranfährt, anschließend seine vorderen Arme nach oben schwenkt und auf dieser ablegt. Danach sollte sich derselbe Vorgang für die beiden hinteren Arme wiederholen. Dieses Modell scheiterte letztendlich aufgrund der mangelnden Stabilität des Baumaterials und der zu geringen Kraft der Motoren, welche nötig gewesen wäre, um den Roboter anzuheben.

3.1.4. Prototyp 4:

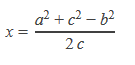
Nachdem wir das Treppensteigen nun nicht mehr fortführten, beschlossen wir uns bei Modell 4 einem Problem zu widmen auf das wir zuvor noch nicht geachtet hatten - die Höhe der Tragfläche. Wir entschieden uns für eine Höhe von 80 cm, damit der Besitzer des Roboters sich nicht bücken muss, um seinen Einkauf zu entnehmen. Dazu entwickelten wir einen Roboter, dessen Fahrgestell dem Prinzip des Gestells des zweiten Prototyps glich, jedoch mit der Veränderung, dass diesmal ein großer Aufbau darauf gesetzt war auf dessen höchsten Punkt sich der Korb für die Einkäufe befand. Außerdem versuchten wir das ersten Mal die Position des Besitzers zu bestimmen. Hierbei setzten wir auf einen Lego Infrarot-Sensor, welcher am Roboter befestigt war und eine dazugehörige Fernbedienung, die beim Besitzer getragen durchgehend Signale senden sollte, anhand derer die ungefähre Richtung und Distanz zwischen Sensor und Sender ermittelt werden kann. Da die dazu nötige Programmierung bereits die Möglichkeiten der Lego-Software überfordern würde, entschieden wir uns auf dem Roboter die auf Linux basierende JavaVM „LeJOS“ zu installieren um nun in Java weiter arbeiten zu können. Mit der Programmierung des Roboters agierte dieser in Ordnung, verlor jedoch oft das Signal der Fernbedienung oder ortete sie an einer vollständig falschen Position. Außerdem war, da weder ein Getriebe noch eine Lenkung eingebaut war, das Fahrverhalten, trotz exakter Software, sehr ungenau.

****

3.1.5. Prototyp 5:

Um die Genauigkeit, welche den 4. Prototypen zum Scheitern gebracht hatte, zu verbessern entschieden wir uns, das seit dem 2. Prototypen nahezu gleich gebliebene Fahrgerüst vollständig neu zu konstruieren und eine Lenkung einzubauen. Außerdem brachten wir nun zwei Infrarot-Sensoren an, welche beide die Distanz zur Fernbedienung bestimmen sollten. Anhand der beiden Distanzen und der Distanz der Sensoren zum Mittelpunkt des Roboters lassen sich mithilfe der Vektorrechnung zwei Schnittpunkte berechnen, wobei der im positiven Bereich liegende Schnittpunkt der Position der Person in einem Koordinatensystem entspricht. Dazu verwendeten wir ein Programm auf Basis der folgenden Terme:







wobei:

a = Radius von Kreis A

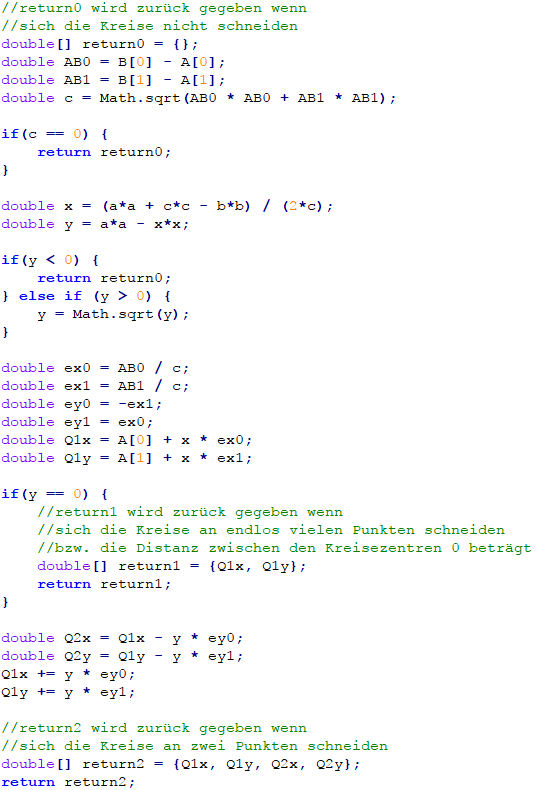
b = Radius von Kreis B

c = Abstand der Kreiszentren

x = X-Abstand der beiden Schnittpunkte zum Ursprung

y = Y-Abstand der beiden Schnittpunkte von der X-Achse

Dieses sah dann aus wie folgt:



Jedoch war nach einigen Testläufen mit der neuen Software klar, dass auch diese aufgrund der zu hohen Ungenauigkeit noch nicht geeignet war um final verwendet zu werden.

3.1.6. Prototyp 6:

Bei unserem sechsten und zum Zeitpunkt des Wettbewerbs aktuellen Prototyp veränderten wir nahezu alles. Die Infrarot-Sensoren wurden von einem Microsoft Kinect Sensor (siehe 3.4.) der zweiten Generation abgelöst. Weiterhin wurde der Aufbau des Roboters massiv verändert, da nun ein Laptop auf ihm befestigt werden musste, um die Daten des Sensors in Echtzeit auswerten zu können. Des Weiteren setzen wir nun auf ein vollständig neues Lenksystem, bei dem die hinteren Räder sich kontinuierlich drehen, während die vorderen Motoren der Lenkung dienen.

Der verwendete Sensor ist in der Lage, durch Ultraschallwellen die vor ihm liegende Umgebung zu scannen und verfügt außerdem über eine Kamera. Mit dem Sensor ist es möglich, eine Person zu verfolgen, indem die Bilddaten nach Personen durchsucht werden (genaueres: siehe ebenfalls 3.4.). Er ist über ein USB-Kabel mit dem angebrachten Laptop verbunden. Durch eine dauerhafte Datenübertragung via WLAN steht der Laptop mit den anderen Geräten in Kontakt.

**3.2. Hardware und Software des EV3**

3.2.1. Die Hardware

Der EV3, ein kleiner Computer hergestellt von LEGO, stellte sich für unser Projekt als durchaus passend heraus. Mit seinem Set verfügt der Besitzer über eine vielfältige Auswahl von sowohl Motoren, welche wir zu der Fortbewegung des Roboters nutzten, als auch Sensoren, welche nur bei Prototyp 1-5 eine Rolle spielten. Durch diese Komponenten war eine gute Grundlage geschaffen.

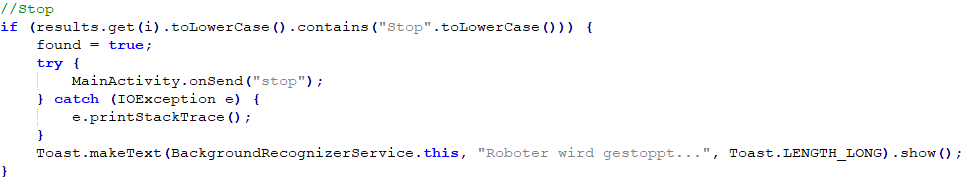
3.2.2. Die Software

Das Manipulieren der Motoren und Sensoren sowie ein sicherer Datenaustausch zwischen dem EV3 und weiteren Geräten wurden möglich, indem wir die jeweiligen Programme in der Programmiersprache Java schrieben. Zwar war schon eine vorgefertigte, graphische Oberfläche zu der Programmierung speziell für EV3s vorhanden, jedoch ist diese nicht auf größere Projekte mit mehreren verbundenen Geräten ausgelegt, weshalb wir damit nicht arbeiten konnten. Um geschriebenen Code auch verarbeiten zu können, wurde ein neues Betriebssystem namens „LeJOS“ auf dem EV3 installiert. Dabei handelt es sich um eine „Java Virtual Machine“, welche durch Linux agiert und extra für den EV3 vorgesehen ist.

Die implementierte Bibliothek „leJOS EV3“ fügt die nötigen Funktionen und Klassen in Java ein, welche beispielsweise das ansteuern der Motoren ermöglichen.

**3.3. Die Applikation**

Ein Grundkonzept unseres Projekts ist die einfache Handhabung des Roboters auch für Menschen, welche sich nicht ausreichend mit Computern auskennen. Daher entwickelten wir eine Applikation für Smartphones, welche Sprachbefehle erkennt und an den Roboter weitergibt. Dieser sollte dem Befehl entsprechend reagieren. Dazu nutzten wir die im Android System integrierte Sprachverarbeitung und ließen diese nach einigen „Hotwords“ suchen. Hier beispielsweise nach dem Wort „Stop“:



3.3.1 Verbindung von Smartphone und EV3

Um die erkannten Begriffe an den EV3 weiterzugeben, richteten wir eine Verbindung via Bluetooth zwischen dem EV3 und dem Handy ein, welche jedes Mal, wenn die Applikation gestartet wird, neu initialisiert wird. Bisher zur Steuerung des Roboters nutzbare Begriffe sind:

* „Stop“
* Der Roboter wird gestoppt
* „Weiter“
* Der Roboter fährt weiter

Die Befehle zum Steuern des Roboters sind auch durch Buttons auf der Startseite der Applikation ausführbar.

3.3.2. Weitere Funktionen der Applikation

Nachdem nun ohnehin eine Applikation auf dem Smartphone des Benutzers installiert werden muss, beschlossen wir diese mit weiteren Funktionalitäten auszustatten.

Mit der Spracherkennung als Grundlage integrierten wir eine Einkaufsliste, welche vollständig durch Sprache gesteuert und bei Bedarf vorgelesen werden kann. Die Sprachsteuerung inkludiert zu diesem Zeitpunkt das Hinzufügen und Entfernen von Produkten, das Leeren der Liste und das vollständige Vorlesen der Liste. Diese Funktionen sind ebenfalls über die graphische Oberfläche der App verfügbar.

Ebenfalls auf Basis der Spracherkennung entwickelten wir eine Notruffunktion, welche, nachdem sie in den Einstellungen in der App konfiguriert wurde, bei einer dabei festgelegten Phrase eine SMS an eine ebenfalls dort definierte Nummer sendet.



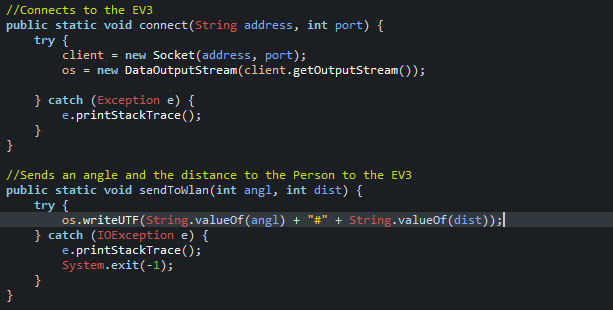
**3.4. Einbindung des Kinect v2 Sensors**

**3.5. Verbindung von Laptop und EV3**

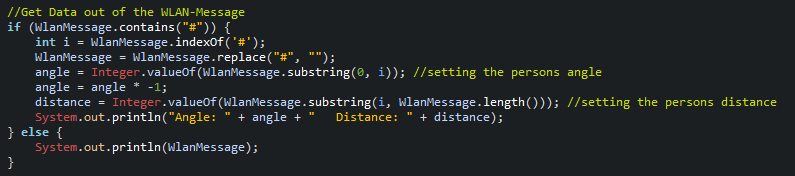
Aufgrund der Tatsache, dass die durch den Laptop empfangenen Daten des Kinect v2 Sensors an den EV3 weitergeleitet werden müssen, damit dieser mit dieses Ergebnissen arbeiten kann, war es nötig eine Verbindung exakt für diese Daten zu erstellen. Dieser Datenfluss ist nur durch ein gemeinsames LAN effektiv, da die Bluetooth-Schnittstelle des EV3 schon für die Verbindung mit dem Handy genutzt wird. Das Netzwerk stellt ein mobiler Hotspot von dem Smartphone dar, damit der Roboter überall genutzt werden kann.

Der Ablauf des Sender-Programms auf dem Laptop sieht wie folgt aus:

Vorerst verbindet sich der Computer, welcher an den Kinect Sensor angeschlossen ist, mit dem EV3 über dessen IP-Adresse. Anschließend werden sowohl der Socket für den Socket als Schnittstelle für die Verbindung, als auch ein „DataOutputStream“ kreiert. Mithilfe der „sendToWlan“ Methode werden nun die erforderlichen Daten über das Netzwerk an den EV3 gesendet.



Auf der Seite des Empfänger wird folglich ein ebenso ein Socket, hier ein „ServerSocket“, erstellt und als Gegenstück zu dem „OutputStream“ ein „InputStream“. Dieser empfängt die vom Laptop gesendeten Daten und liest diese aus.



Durch das auslesen ist der Roboter nun über die exakte Position der Person vor ihm informiert und steuert dementsprechend seine angeschlossenen Motoren in dessen Richtung.

4. ERGEBNISSE

Nachdem nun die Funktionsweise und die Idee unseres Projekts erläutert wurden, möchten wir einen realistischen Anwendungsfall nennen, indem wir den von uns entwickelten Roboter mit einer Haushilfe vergleichen. Dabei achten wir achten wir sowohl auf die Kosten, als auch auf andere Vorteile sowie Nachteile.

Für den Vergleich gingen wir von einem durchschnittlichen Lohn für die Haushaltshilfe aus, nämlich 10€ auf eine Stunde Arbeitszeit gerechnet. Bei schon bei nur drei Stunden Arbeitszeit in einer Woche ergibt sich auf einen Monat hochgerechnet ein Betrag von 120€. In einem Jahr wäre folglich eine Summe von 1560€ zu begleichen.

Da es sich bei unserem Roboter ausschließlich um einen Prototyp handelt, würden Produktionskosten im Falle einer Massenproduktion eindeutig sinken. Der derzeitige Roboter hat einen Wert von ca. 1300€. Den größten Wert am Roboter stellt der installierte Laptop dar. In einer Massenproduktion würde ein solcher Bildschirm, wie er am Laptop vorhanden ist, wegfallen. Viele weitere Details des Laptops sind ebenso nicht brauchbar (Webcam, etc.). Auch der Preis des Gestells und der Motoren sowie Sensoren würden sinken.

Wir schlussfolgern einen Wert von 800€.

**(Graph einfügen)**

Wie erkennbar ist, lohnt es sich für den Nutzer schon nach **X** Wochen, diesen Roboter zu nutzen.

**(weiterschreiben)**

Der Source-Code der Programme ist unter <https://github.com/>(...) zu finden.

**Aufzählung der genutzten Libraries?**

5. DISKUSSION

**5.1. Probleme bei der Umsetzung**

Um unser Projekt effektiv zu gestalten, damit es wirklich einfach zu bedienen ist und einen Wert hat, entschieden wir uns mit der Programmiersprache Java zu arbeiten, damit wir mit dieser die einzelnen Computer programmieren können. Da wir vor dem Projekt keinerlei Vorkenntnisse mit irgendeiner Programmiersprache hatten, war es nötig jene Programmiersprache neu zu lernen.

Dies erforderte viel Zeit und war nicht selten schwierig zu verstehen.

Oft kam es auch zu Schwierigkeiten bei der Umsetzung des Programms durch den Roboter, da die Umwelt, in welcher sich der Roboter bewegt, viele Probleme darbietet, wie beispielsweise **…()**

**5.2. Weiterentwicklungsmöglichkeiten**

Im Bezug auf die Weiterentwicklung des Projekts haben wir viele Ideen. Die Umsetzung jedoch gelang nicht aufgrund zu hoher Kosten durch Material.

Eine der Ideen liegt darin, von unserem jetzigen Fahrgestell abzuweichen aufgrund der zu hohen Instabilität und den Roboter auf ein „Turtlebot“ (http://www.turtlebot.com) Gestell zu verlagern. Dieses weißt sowohl eine große Stabilität, als auch starke Motoren auf, welche sich optimal für unsere Nutzung darstellen.

Außerdem enthält der „Turtlebot“ mehrere leistungsstarke Akkus, mit denen die benötigte Energie der Motoren abgedeckt würde.

Eine weitere Methode, um das Projekt zu verbessern, ist die Ermöglichung des Roboters Treppen steigen zu können. Zwar scheiterte unser Versuch diese umzusetzen, jedoch erscheint es uns möglich mit passendem Material eine solch hilfreiche Funktion zu kreieren. Dazu benötigt es nämlich vor allem leistungsstarke Motoren und die passende Mechanik.

**(Optimierung des Programms, sodass es auch auf leistungsschwächeren Computern laufen kann)**

**(Idee GPU)**

6. DANKSAGUNG

Wir danken niemandem, außer meiner Oma.

7. QUELLEN- UND LITERATURVERZEICHNIS

„Android Tutorials Deutsch“, The Morpheus Tutorials:

<https://www.youtube.com/playlist?list=PLNmsVeXQZj7qShNeVpdDAQedIq2n8BvqC>, letzter Aufruf: 17.12.2017

**(Google Cloud Speech)**