

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. Notruffunktion

**3.4. Ausführliche Einbindung des Kinect v2 Sensors**

**3.5. Verbindung von Laptop und EV3**

4. MÖGLICHE FEATURES

5. ERGEBNISSE

6. DISKUSSION

**6.1. Probleme bei der Umsetzung**

**6.2. Weiterentwicklungsmöglichkeiten**

7. DANKSAGUNG

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. Dadurch können vor allem alleinlebende Menschen sich nicht mehr selbst versorgen und sind auf Unterstützung durch Haushaltshilfen angewiesen, welche viel Geld kosten, das man selten aufbringen kann.

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

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

Um das Problem, welches das Tragen von schweren Einkäufen o.ä. darstellt, zu lösen, entschieden wir uns einen Roboter zu konstruieren welcher diese Aufgabe übernehmen sollte. Dieser sollte, um für die meist Technik-Unerfahrene ältere Generation leicht zu handhaben sein, autonom, somit ohne weitere manuelle Steuerung, seinem Besitzer folgen und ansonsten nur mit Sprachbefehlen gesteuert werden. Außerdem muss der Roboter sich den Gegebenheiten, wie zum Beispiel der unter Umständen geringen Geschwindigkeit älterer Menschen anpassen können. Dazu kommen noch einige weitere Umstände, die es zu beachten gab während der Planung für den Bau eines solchen Helfers, wie beispielsweise eine geeignete Höhe der Tragefläche, um ein anstrengendes Bücken des Besitzers vermeiden zu können.

2. ART UND WEISE DES VORGEHENS

Um allen Bedingungen gerecht zu werden, die für einen guten Roboter nötig sind, planten wir eine ganze Reihe von verschiedenen Modellen, wobei die verwendete Technik und Software sich, auf Basis der vorangehenden Modelle, immer weiter verbessern sollte.

Durch dieses System setzten die ersten Prototypen noch auf sehr einfache Technik sowie Mechanik und es fehlten viele 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 Methode entschieden.

Mit der Zeit sollten außerdem noch einige weitere, zusätzlich Funktionen hinzukommen welche den Nutzern des Roboters ihren Alltag weiter erleichtern sollen.

Nach dem Bau eines Prototyps erstellten wir ein passendes Programm, immer an die Fähigkeiten des gegenwärtigen Roboters angepasst. Somit teilten wir das gesamte Projekt in mehrere Themenbereiche ein, um sowohl einen besseren Überblick über das Projekt zu beizubehalten, als auch die Konzentration auf jedes einzelne Problem zu richten. Dadurch wurde eine effektive Arbeitsweise geschaffen.

3. UMSETZUNG

**3.1. Das Entwickeln eines Roboters**

Bei der Entwicklung der Prototypen mit dem Ziel des finalen Prototyps gingen wir schrittweise vor, was heißt, dass wir mit jeder neuen Entwicklungsstufe ein einzelnes oder zwei Probleme lösten, um dadurch ein übersichtliches und nachvollziehbares Projekt zu kreieren. Da wir beide in Besitz von vielen LEGO-Technik-Bausteinen sind und unser Budget für neue Bausteine, zum Beispiel robustere, begrenzt und nicht genügend war, fokussierten wir uns auf den Bau der Roboter mithilfe der LEGO-Bausteine.

Die genutzten Bauteile haben jedoch auch den Vorteil, dass sie allesamt miteinander verknüpfbar sind. Dieser Eigenschaft bedienten wir uns und waren somit in der Lage beträchtliche und effektive Objekte in Form der Prototypen zu erschaffen.

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 kaum 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 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 kam es erstmals zu dem Versuch, den Roboter Treppen steigen zu lassen. Wir setzen bei der Umsetzung auf ein System, bei dem sich der Roboter durch vier an den Seiten angebrachte Arme stemmt. Das dafür entwickelte Programm sollte dafür sorgen, dass der Roboter sich mit seinen Rädern auf eine nächste Treppenstufe vorschiebt und dadurch auf sie gelangt. Die Arme waren in der Lage sich zu heben und zu senken, nämlich durch die aufgebrachte Motorkraft, welche über mehrere Nylonfäden auf die Arme übertragen wurde. **(Weiterschreiben, habe gerade eine Gedankenblokade)**

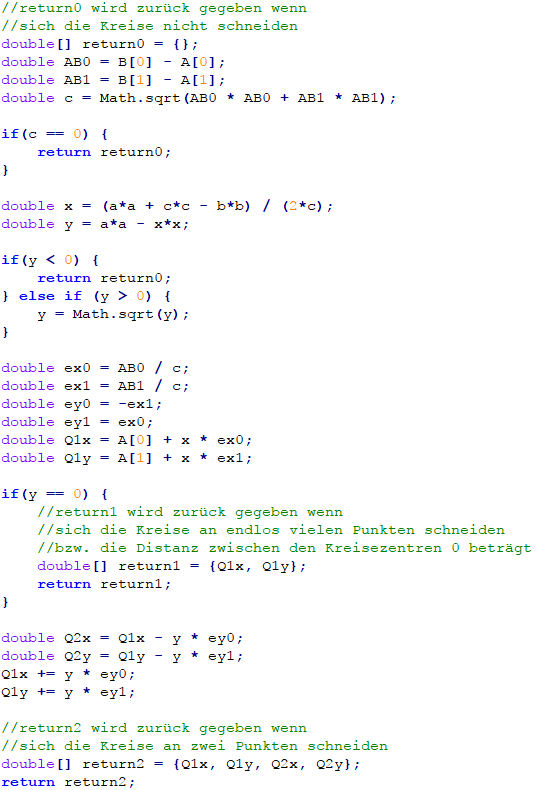
3.1.4. Prototyp 4:

Nachdem wir das Treppensteigen nun nicht mehr fortführten, beschlossen wir uns bei Modell 4 einem Problem zu widmen auf das wir zuvor noch gar nicht geachtet hatten - die Höhe. Wir entschieden uns für eine Höhe von 80 cm, damit ein Besitzer des Roboters sich nicht bücken muss, um seinen Besitz zu erreichen und entwickelten einen Roboter, dessen Fahrgestell dem Prinzip des zweiten Prototyp glich, jedoch mit der Veränderung, dass diesmal ein großer Aufbau darauf gesetzt war auf dessen Spitze sich der Korb für die Einkäufe befand. Außerdem verwendeten wir das erste Mal eine Technik um die Position des Besitzers zu bestimmen. Hierbei setzten wir auf einen Lego Infrarot-Sensor, welcher am Roboter befestigt war und eine dazugehörige Fernbedienung, die vom Besitzer aus durchgehend Signale senden sollte, anhand deren die ungefähre Richtung und Distanz zwischen Sensor und Sender ermittelt werden kann. Da die Programmierung dazu bereits die Möglichkeiten der Lego-Software überfordern würden, entschieden wir uns auf dem Roboter die auf Linux basierende JavaVM „LeJOS“ auf dem EV3 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 planen und eine Lenkung einzubauen. Außerdem brachten wir nun zwei Infrarot-Sensoren an, welche beide die nur 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 entspricht. Dazu verwendeten wir ein Programm auf Basis der folgenden Terme:



x =

y =

Jedoch war nach einem ersten Testdurchlauf mit der neuen Software erneut ein Problem bei der Genauigkeit der Messwerte zu eruieren. Nach einigen Testdurchläufen, um jene Werte der Distanz zu Fernbedienung genau zu ermitteln, wurde klar, dass der Infrarotsensor wiederholt ungenaue und teils falsche Werte ausgab.

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 3.5. Ausführliche Einbindung des Kinect v2 Sensors) der zweiten Generation. Weiterhin wurde der Aufbau des Roboters intensiv verändert dadurch, dass nun ein Laptop auf ihm befestigt werden musste, um die enormen Datenströme des Kinect v2 Sensors zu verarbeiten. Der Sensor ist in der Lage, durch Infrarotstrahlung die vor ihm liegende Umgebung zu scannen und verfügt noch dazu über eine Kamera. Mit dem Sensor ist möglich, eine Person dauerhaft zu markieren, wodurch das Folgen des Roboters ermöglicht wird. Der Sensor ist über ein USB-Kabel mit einem Laptop verbunden. Durch eine dauerhafte Datenübertragung via einer WLAN Verbindung ist der Laptop in das System des Roboters implementiert.

**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, die in unserem Fall zur besseren Detektion der Umgebung verwendet wurden. Durch das Benutzen dieser Komponenten war eine gute Voraussetzung geschaffen. Wir nutzen nun zwei Motoren für den Antrieb des Konstrukts und zwei weitere um den Roboter lenken lassen zu können. Somit sind alle Ports für Motoren an dem kleinen Computer besetzt. (…Sensoren)

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 weißt sie immer noch einige Fehler auf, sodass wir keinesfalls effektiv damit 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“ fügte die nötigen Kommandos in Java ein, welche beispielsweise ein Stoppen oder das Starten angeschlossener Motoren erzwangen.

(Ausschnitt vom Programm)

**3.3. Die Applikation**

Nun war eine praktische Grundlage für das Projekt geschaffen, jedoch fehlte unserer Meinung nach noch die simple Bedienung des Roboters für Menschen, welche körperlich eingeschränkt sind oder sich weniger gut mit Computern auskennen. Die Lösung durch simple Handhabung stellte eindeutig eine passende App dar, denn eine solche verfügt unter anderem über eine graphische Oberfläche, die leicht zu bedienen ist.

Als eine simple Nutzung des Roboters via eine Applikation beurteilten wir eine Steuerung durch Sprache als effektivste Art. Die Applikation sollte unseren Vorstellungen nach über einen „BackgroundRecognizerService“ verfügen, damit ein Sprachbefehl auch dann erkannt und ausgewertet wird, wenn der Bildschirm des Smartphone nicht angeschaltet ist. (noch nicht fertig)

3.3.1 Verbindung von Smartphone und EV3

Nun, da eine Spracherkennung in der App auf einem Smartphone eingebaut worden war, müssen die erkannten und ausgewerteten Daten an den Computer des Roboters, den EV3, gesendet werden. Die Verbindung zwischen den beiden Geräten wird via Bluetooth erstellt, sobald die App den richtigen EV3 (wodurch?) erkennt, wird eine Verknüpfung hergestellt. Dazu wird vorerst ein „BluetoothSocket“ erstellt, um sich mit dem vorgegebenen Device verbinden zu können. Mithilfe der „switch-case“ Methode werden nun erfasste Sprachbefehle an den EV3 versendet.

Folgende Sprachbefehle sind nun aufrufbereit, sobald sie von der App erkannt werden:

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

Noch hinzu befestigten wir die Aufbewahrungskiste auf dem Roboter, welche beispielsweise Einkäufe tragen soll. Um deren Schutzklappe öffnen zu lassen, kann man sich ebenfalls Sprachbefehlen bedienen:

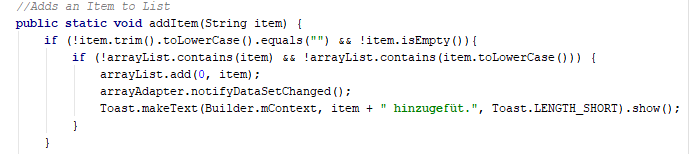
* „Klappe öffnen“
* Die Aufbewahrungskiste wird geöffnet
* „Klappe schließen“
* Die Aufbewahrungskiste wird geschlossen

Die Befehle zum Steuern der Fortbewegung des Roboters, also Stoppen und Fortfahren der Motoren, sind auch in der MainActivity der Applikation durch Buttons ausführbar.

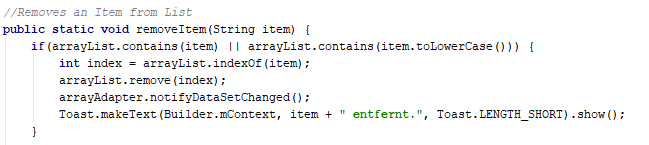
Als ein Feature wurde in die Applikation eine Einkaufsliste hinzugefügt. Diese ist ebenfalls durch Sprachbefehle zu steuern, jedoch ist es auch möglich, dass man seine Einkäufe via Tastatur eingibt.

(Bild der MainActivity)

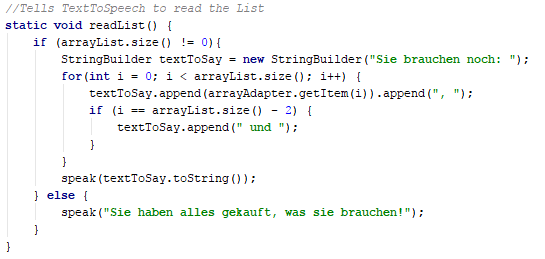
Jene Funktion ermöglicht nun, dass durch das Erwähnen des benötigten Gegenstands gefolgt von dem Wort „hinzufügen“, das Genannte in eine Liste addiert wird.



Es ist auch möglich die in der Liste aufgeführten Gegenstände zu entfernen, nämlich durch langes Tippen auf das Item oder das Aufsagen des zu entfernen gewünschten Begriff gefolgt von den Wort „entfernen“.



Weiterhin kann man die Liste durch den Sprachbefehl „Liste vorlesen“ vorlesen lassen. Diese Funktion wird durch die „TextToSpeech“ Methode ermöglicht, welche über die Server von Google läuft. Ebenso wird die Spracherkennung durch Google ausgewertet. Folglich ist eine Internetverbindung nicht verzichtbar.



3.3.2. Notruffunktion

Bei der Notruffunktion handelt es sich um eine SMS-Nachricht, welche der Besitzer der Applikation in einem Notfall an eine beliebige Handynummer senden kann. Ausgesprochen nutzbringend ist eine derartige Funktion zum Beispiel für ältere Menschen, die sich aufgrund ihrer körperlichen Eigenschaften schneller und garvierender verletzen können.

Diese Notfall-Nachricht wird abgesendet, sobald ein bestimmter, vom Besitzer der Applikation festgelegter Ausdruck, durch Sprache erwähnt wird. Auch der Inhalt der Nachricht wird von dem Besitzer festgelegt.

(Screenshot von der Activity)

(Screenshot vom entsprechenden Code)

**3.4. Ausführliche Einbindung des Kinect v2 Sensors**

Der Kinect v2 Sensor (<https://developer.microsoft.com/de-de/windows/kinect/>) bot uns die Gelegenheit, Menschen auf dem Bildschirm eines Computers abbilden zu lassen. Man hat sowohl Zugriff auf einen Bilderstream der eingebauten Kamera, als auch auf eine Tiefenkarte, erstellt durch die Detektion der Umgebung mithilfe von Infrarotstrahlung.

**(Beispiel einer Tiefenkarte, Foto)**

Das Ziel der Nutzung des Kinect-Sensors bestand für uns darin, die gewünschte Person möglichst exakt zu orten. Ein weiteres Ziel bestand darin, den Computer zwischen verschiedenen Personen unterscheiden lassen zu können, beispielsweise anhand der Farbe der getragenen Kleidung, um somit immer einem Besitzer folgen zu können.

In der Theorie betrachtet ist das dafür geschriebene Programm in zwei Teile eingeteilt.

In einem ersten Teil soll eine beliebige Anzahl an Bildern von dem Besitzer fotografiert und gespeichert werden. Die Fotos, aufgenommen von der Kamera des Sensors, sollen den Körper des Besitzers von möglichst unterschiedlichen Seiten darstellen.

In dem zweiten Teil, welcher in einer Endlosschleife ausgeführt wird, empfängt der Computer durch den Sensor in Echtzeit und gleicht diese mit jenen Bildern ab, welche er zuvor gemacht hatte.

Dabei sucht das Programm nach den meisten Ähnlichkeiten zwischen den Bilder und kann bei einem erfolgreichem Abgleich schlussfolgern, dass es sich um den Besitzer handelt. Anschließend werden die Positionsdaten ermittelt.

**(praktischer Ablauf + Screenshots)**

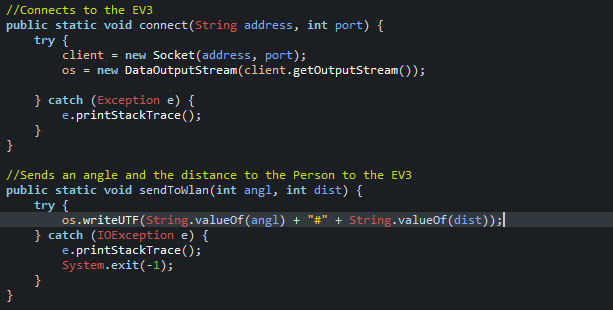
* Probleme der langsamen Verarbeitung durch die CPU, Idee GPU

**3.5. Verbindung von Laptop und EV3**

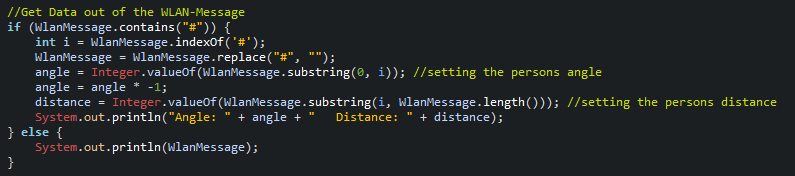
Aufgrund der Tatsache, dass die durch den Laptop empfangenen Daten des Kinect v2 Sensors an den EV3 weitergeleitet werden müssen, damit dieser mit dieses Ergebnissen arbeiten kann, war es nötig eine Verbindung exakt für diese Daten zu erstellen. Dieser Datenfluss ist nur durch ein gemeinsames LAN effektiv, 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. MÖGLICHE FEATURES

(möglicher Ausbau des Roboters, um ihn stabiler zu machen)

5. ERGEBNISSE

(einfach nur eine Zusammenfassung?)

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

Aufzählung der genutzten Libraries?

6. DISKUSSION

**6.1. 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.

**6.2. 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.

DANKSAGUNG

Wir danken niemandem, außer meiner Oma.