Fachhochschule Hagenberg

Softwareprojekt-Engineering, SS 2025

Pflichtenheft - LEGO Roboter Projekt

Pflichtenheft der Projektgruppe G12 im 2. Semester

Gruppe 12

Autor
Tim Peko

Korrekturleser
Moritz Kieselbach

Teammitglieder
Moritz Kieselbach
Tim Wahlmüller
Tim Peko
Alexander Kranl
Alexandra Usuanlele

Version 2.0

Inhaltsverzeichnis

Einleitung . . . 3
1 Einführung in das Projekt . . . 4
1.1 Stakeholder . . . 4
1.2 Zeitlicher Rahmen . . . 4
2 Ausgangssituation (Istzustand) . . . 5
2.1 Lego-Mindstorms-Roboter . . . 5
3 Aufgabenstellung (Sollzustand) . . . 6
3.1 Grundanforderungen . . . 6
3.2 Rahmenbedingungen . . . 7
4 Schnittstellenanforderungen . . . 9
4.1 Hardware-Umgebung Schnittstelle . . . 9
4.2 Hardware-Benutzer Schnittstelle . . . 9
4.3 Hardware-Software Schnittstelle . . . 9
5 Systemanforderungen . . . 10
5.1 Hardware . . . 10
5.2 Software . . . 10
5.3 Implementierung . . . 10
6 Betriebsanforderungen . . . 11
6.1 Bedienung . . . 11
6.2 Untergrund . . . 11
6.3 Beleuchtung . . . 11
6.4 Betriebsdauer . . . 11
6.5 Fehlerbehandlung . . . 11
7 Qualitätsanforderungen . . . 13
7.1 Genauigkeit . . . 13
7.2 Geschwindigkeit . . . 13
7.3 Zuverlässigkeit . . . 13
7.4 Sicherheit . . . 13
7.5 Ressourceneffizienz . . . 13
8 Anforderungen an die Projektentwicklung . . . 14
8.1 Entwicklungsmethodik . . . 14
8.2 Dokumentation . . . 14
8.3 Tests . . . 14
8.4 Kommunikationswege . . . 14
9 Abnahmekriterien . . . 16
9.1 Dokumente . . . 16
9.2 Wettbewerb bzw. Testtage . . . 16
10 Design . . . 18
10.1 Hardware . . . 18
10.2 Programmierung . . . 19
10.3 Ablaufplan . . . 19
Anhang . . . 21
Teammitglieder der Gruppe G12 . . . 21
Begriffe . . . 21

Einleitung

Im Rahmen des Sommersemesters 2025 wird im Modul Software-Projekt Engineering 2 ein Projekt durchgeführt. Dieses Projekt dient als Übung, um die Kursteilnehmer mit klassischem Projektmanagement für Softwareprojekte vertraut zu machen und sie auf organisatorische Strukturen und Konventionen in der freien Wirtschaft vorzubereiten.

Das ***Projektteam*** in dieser Projektausführung ist die Gruppe **G12**, deren **Partnergruppe** G11 ist. Die **Partnergruppe**n müssen sich Ressourcen teilen. In diesem Projekt wird das ein Lego-Mindstorms-Roboter sein. Unter Industriebedingungen könnte dies ein größerer Rechenserver sein, dessen Ressourcen auf mehrere Teams verteilt werden müssen.

Die konkreten Teammitglieder des ***Projektteam***s **G12** werden im Anhang angeführt.

1 Einführung in das Projekt

Dieses Projekt umfasst die Entwicklung eines Lego-Mindstorms-Roboters, der in der Lage sein muss, Lego-Blöcke von einem erhöhten Steinebereich auf einen niedrigeren Zielbereich zu transportieren und dabei nach Farben sortiert zu stapeln. Ort und Platzierung der Blöcke auf dem Zielbereich sind irrelevant. Der Roboter muss autonom arbeiten und seine Aufgabe ohne menschliches Eingreifen erfüllen können.

1.1 *Stakeholder*

Im Rahmen dieses Projekts wurden folgende Stakeholder identifiziert:

1.1.1 Kursleiter FH-Prof. DI Dr. Herwig Mayr

Der **Kursleiter** FH-Prof. DI Dr. Herwig Mayr stellt die Aufgabenstellung, die Ziele des Projekts sowie die benötigten Ressourcen und Rahmenbedingungen bereit. Er fungiert zugleich als Auftraggeber des Projekts.

1.1.2 Partnergruppe G11

Die **Partnergruppe** G11 arbeitet parallel an derselben Aufgabenstellung und teilt die Ressourcen mit der Gruppe **G12**. Dies erfordert eine enge Kooperation zur gemeinsamen Nutzung des Lego-Mindstorms-Roboters und eine sorgfältige Abstimmung der Arbeitszeiten und -abläufe zwischen den Gruppen.

Projektleiterin: Madlen Moldaschl

1.1.3 Koordinationsgruppe

Die **Koordinationsgruppe** vertritt die Gesamtheit aller Projektgruppen und fungiert als offizielle Schnittstelle zwischen diesen und dem **Kursleiter**. Sie definiert in etwa Bewertungskriterien, genaue Umgebungsmaße oder spricht allgemeine Probleme mit dem **Kursleiter** an.

Vertreterin der G12: Alexandra Usuanlele

1.2 Zeitlicher Rahmen

Das Projekt folgt einem strukturierten Zeitplan mit drei definierten Meilensteinen, auch Meilensteintage genannt:

- 2025-05-08: Der erste Meilenstein am 08.05.2025 markiert die Fertigstellung einer lauffähigen Alpha-Version des Roboters. Diese Version sollte die grundlegende Funktionalität demonstrieren, muss aber noch nicht alle Anforderungen vollständig erfüllen.
- 2025-05-22: Der zweite Meilenstein am 22.05.2025 ist der Tag der Fertigstellung einer Beta-Version, die bereits alle funktionalen Anforderungen erfüllt, aber noch Optimierungsbedarf bei nicht-funktionalen Anforderungen haben kann.
- 2025-06-18: Der dritte und finale Meilenstein am 18.06.2025 ist der Tag des finalen Wettbewerbs und zugleich der Abgabetermin für das Projekt. Zu diesem Zeitpunkt muss das Projekt vollständig fertiggestellt sein und alle funktionalen sowie nicht-funktionalen Anforderungen erfüllen.

2 Ausgangssituation (Istzustand)

Der für das Projekt zu verwendende Lego-Mindstorms-Roboter wird aus einem vergangenen Projekt übernommen. Seine Grundkonfiguration war dadurch bereits vorhanden und wird nachfolgend genauer beschrieben. Der Roboter muss jedoch konkret an dieses Projekt angepasst werden. Die bereits vorhandene Grundkonfiguration darf - muss aber nicht - erhalten bleiben.

2.1 Lego-Mindstorms-Roboter

Der Roboter wird übernommen mit:

- Einem groben Fahrwerk
- Einem losen Greifarm
- Einem betriebsfähigen EV3-Controller
- Mehreren losen Lego-Technik-Teilen
- Mehreren Batterien
- Unterschiedlichen Sensoren/Motoren

3 Aufgabenstellung (Sollzustand)

Es ist eine automatisierte Lösung für das Stapeln von Lego-Steinen nach Farbe zu entwickeln. Der zu entwickelnde Roboter muss in der Lage sein, Lego-Blöcke anhand ihrer Farbe zu erkennen, diese von einem erhöhten Steinebereich zu greifen und auf einem Zielbereich nach Farben sortiert zu stapeln. Diese Aufgabe muss autonom erledigt werden.

Der Umgebungsaufbau sowie sonstige Gegebenheiten werden im Folgenden detailliert beschrieben. Im Abschnitt 10 wird das Design Konzept der Projektgruppe beschrieben.

3.1 Grundanforderungen

3.1.1 Stapel

Die gebildeten Stapel dürfen nur genau eine Farbe beinhalten. Sie müssen stabil stehen bleiben, ansonsten gibt es keine Präzisionsanforderungen. Eine maximale Stapelhöhe ist nicht vorgegeben, wobei ein möglichst hoher Stapel anzustreben ist.

Messgröße	Wertvorgabe
Höhe	Minimal: 1 Block Maximal: **unbegrenzt**
Abstand zu anderen Stapeln	Minimal: 0 cm
Grundfläche	Genau: 1x1 Block
Anzahl Farben	Genau: 1

Tabelle 1: Stapeldimensionen

Die einzelnen Blöcke eines Stapels dürfen in jeglicher Rotation platziert werden. Ein von der Arbeitsfläche heruntergefallener Block gilt als verloren und darf nicht wieder aufgehoben werden.

3.1.2 Aufbau

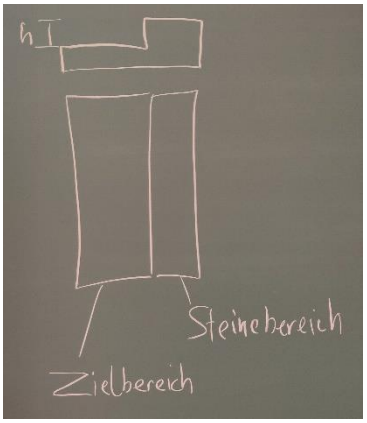

Abbildung 1: Skizze des Umgebungsaufbaus vom **Kursleiter**

3.1.2.a *Steinebereich*

Die Lego-Blöcke befinden sich auf einem erhöhten, ebenen, geraden, rechteckigen Steinebereich (z.B. ein Holzbalken), der auf einem Holztisch platziert ist.

Die Blöcke liegen in einer Reihe auf dem *Steinebereich* und haben zum Rand und zu anderen Steine einen Abstand von 20cm. Die Farbe der Blöcke ist zufällig und folgt keinem Muster. Jedoch ist sie für alle Projektgruppen gleich.

Dicker Holzbalken mit breitem Balken am Boden:
- Vorteile: stabiler, weniger Risiko zu fallen.

3.1.2.b *Zielbereich*

Der rechteckige Zielbereich befindet sich neben dem Steinebereich auf demselben Untergrund, auf dem sich auch der Roboter befindet. Er ist also auch gleichzeitig die Arbeitsfläche und wird so groß gewählt, dass genügend Platz für den Roboter und die zu bildenden Stapel ist.

Es ist zu beachten, dass bei größeren *Arbeitsfläche*n möglicherweise mehrere Tische zusammengestellt werden, was zu leichten Unebenheiten führen kann.

3.1.2.c Sonstiges

Folgende Aspekte werden am Tag vorm *Alpha-Testtag* festgelegt:

- Beschaffenheit des Tisches
- Dimensionen des Tisches, Steinebereich und Zielbereich
- Höhendifferenz zwischen Steinebereich und Zielbereich

3.1.3 Markierungen

Es gibt keine Markierungen auf dem Tisch, die der Roboter als Orientierung verwenden könnte. Diese dürfen auch nicht vom ***Projektteam*** angebracht werden.

3.2 Rahmenbedingungen

Die folgenden Rahmenbedingungen sind für die Durchführung des Projekts festgelegt. Nicht explizit definierte Bedingungen können von der Projektgruppe in Absprache mit dem **Kursleiter** selbst festgelegt werden.

3.2.1 Hardwarekomponenten

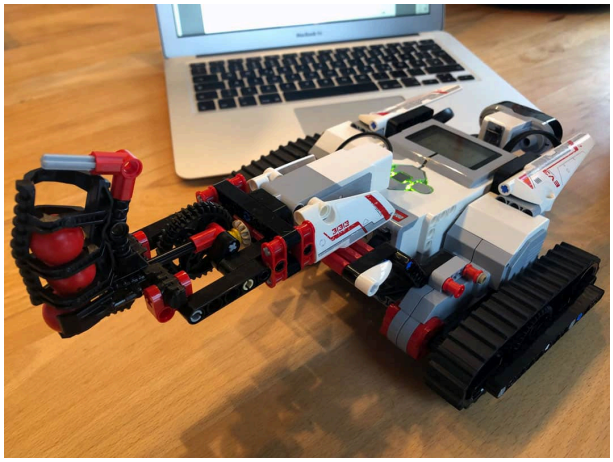

Abbildung 2: Reale Darstellung eines Lego-Mindstorms-Roboters

Als Basis für den Roboter dient ein Lego-Mindstorms-EV3-Mono-Brick, der mit Batterien betrieben wird. Folgende *Peripheriegerät*e sind ebenfalls verfügbar:

- 3x Motoren
- 1x Ultraschallsensor
- 2x Farbsensoren
  - *NXT RGB-Sensor*
  - *Neu RGB-Sensor*
- 1x Gyrosensor

Zusätzlich steht ein Lego-Technik-Set zur Verfügung, das für den Aufbau des Roboters genutzt werden kann.

Das Hinzufügen neuer Teile oder Komponenten ist nicht erlaubt. Der **Kursleiter** bietet auf Anfrage die Möglichkeit, kaputte oder fehlende Teile zu ersetzen.

4 Schnittstellenanforderungen

Dieses Kapitel beschreibt die Schnittstellen, die der Roboter unterstützen muss, um seine Aufgabe erfüllen zu können.

4.1 Hardware-Umgebung Schnittstelle

Der Roboter muss physisch mit seiner Umgebung interagieren können. Die Konstruktion des Roboters muss robust genug sein, um seine Aufgaben zuverlässig ausführen zu können, auch wenn leichte Variationen in der Umgebung oder den Eigenschaften der Blöcke auftreten.

4.1.1 Lego-Blöcke

Die Lego-Blöcke haben die gleiche Form und Größe.

Größe	Wert
Höhe	~3,15 cm
Breite	~3,3 cm
Länge	~3,3 cm

Tabelle 2: Lego-Blöcke Dimensionen

Sie unterscheiden sich nur in der Farbe.

Die konkreten Farben werden am Alpha-Testtag basierend auf der Zuverlässigkeit der Sensoren festgelegt. Es kann von $3 \leq n \leq 10$ Farben ausgegangen werden. Es sind höchstens ~10 Steine pro Durchlauf zu stapeln.

Die Blöcke auf dem Steinebereich sind niemals verdeckt, bedeutet es gibt keine Situation, in der ein Block nur teilweise oder von hinten durch einen anderen Block verdeckt wird.

4.2 Hardware-Benutzer Schnittstelle

Der Roboter ist bereits vorprogrammiert und muss nur noch analog, manuell von einem Benutzer gestartet werden. Danach läuft der Roboter autonom ohne menschlichen Eingriff. Beim Wettbewerb wird das Programm von einem Tutor gestartet.

Der Roboter muss kein Feedback an den Benutzer geben. Es ist keine Statusanzeige oder Ton bei Schlüssel- oder Endereignissen (wie Beendigung des Durchlaufs) zu implementieren.

4.3 Hardware-Software Schnittstelle

Für die Programmierung wird eine Standard-EV3-Umgebung vorausgesetzt, wobei PyBricks oder vergleichbare Programmierumgebungen eingesetzt werden können. Die Wahl der Programmiersprache liegt im Ermessen der Projektgruppe und sollte auf Basis der verfügbaren Kenntnisse und der Eignung für die Aufgabe getroffen werden.

5 Systemanforderungen

Die Systemanforderungen definieren die Hardware- und Softwarevoraussetzungen sowie die an das zu entwickelnde System.

5.1 Hardware

Für die Umsetzung des Projekts wird ein Lego-Mindstorms-EV3-Roboter mit allen notwendigen Komponenten benötigt. Zur Programmierung des Roboters wird benötigt:

1. eine Micro-SD-Karte vom Typ SDHC (4GB - 32GB)
2. einen Laptop für die Entwicklung der Programme
3. ein Mini-USB-Kabel zur Übertragung

	Länge	**Breite**	**Höhe**
Minimum	ausstehend	ausstehend	ausstehend
Maximum	60 cm	30 cm	60 cm

Die Messpunkte dafür am Roboter müssen vom ***Projektteam*** selbst festgelegt und dokumentiert werden. Dazu reicht das *Pflichtenheft*.

5.2 Software

Es wird eine geeignete Entwicklungsumgebung zur Programmierung des Lego-Mindstorms EV3 benötigt.

5.3 Implementierung

Es besteht keine Anforderung an den konkreten Ablauf eines Durchlaufs. Es muss lediglich die grundlegende Anforderung erfüllt werden:

1. Die Lego-Steine werden von einem *Steinebereich* entnommen
2. Sie werden in Form von Stapeln auf einem *Zielbereich* abgelegt
3. Dabei wird nach Farben sortiert

6 Betriebsanforderungen

Die Betriebsanforderungen beschreiben die Bedingungen, unter denen der Roboter seine Aufgabe erfüllen kann, sowie Anforderungen an seine Laufzeit und Fehlerbehandlung.

6.1 Bedienung

Der Roboter muss seine Aufgabe ohne zusätzliche manuelle Eingriffe ausführen können. Davon ausgenommen sind:

- Aufsetzen der *Betriebsumgebung*
- Positionierung des Roboters
- Programmstart

6.2 Untergrund

Der Roboter ist darauf ausgelegt, unter folgenden Untergrundbedingungen zu arbeiten. Die Funktionsfähigkeit muss auf anderen Untergründen nicht gewährleistet werden.

- Holz (Tisch)
- Grundsätzlich eben
  - Kleine Spalten zwischen zusammengeschobenen Tischen sind zulässig
- Horizontal unverformt
  - Nicht konvex
  - Nicht konkav
- Nicht rutschig
- Nicht relevant schief (gemessen nach Wasserwaage)

6.3 Beleuchtung

Die Umgebung wird mittels Tageslicht sowie künstlichem Raumlicht beleuchtet. Die Beleuchtung des Audimax wird so eingestellt, dass eine möglichst hohe Helligkeit erreicht wird. Da die Vorhänge allerdings manchmal geschlossen und geöffnet werden, ist eine gewisse Variation der Beleuchtung zu erwarten.

6.4 Betriebsdauer

Der Roboter muss in der Lage sein, seinen Arbeitsauftrag bzw. einen Durchlauf einmalig vollständig durchzuführen. Es gibt keine Anforderung an den Roboter, mehrere Durchläufe kontinuierlich abarbeiten zu können.

Der Roboter schließt einen Durchlauf in ~3 Minuten ab.

6.5 Fehlerbehandlung

Der Roboter muss tendenziell nicht in der Lage sein, im Fehlerfall automatisch einen Neustart auszuführen. Es muss nicht auf verschiedene Fehlerarten (Prozess-, Umgebungs-, Bedienungsfehler) reagiert werden.

Je nach Fehlerart und -grad ist dies dennoch sinnvoll, um seine Aufgabe erfüllen zu können. Die Implementierung der Fehlerbehandlung erfolgt daher nach den besten Möglichkeiten anhand der Ressourcen des durchführenden ***Projektteam***s.

6.5.1 Typische Fehlerfälle

Zu den typischen Fehlerfällen, die berücksichtigt werden sollten, gehören:

- **Fehlgeschlagene Farberkennung**: Der Roboter kann die Farbe des Blocks nicht erkennen.

  - **Vorgabe**: Jeder Block ist eindeutig erkennbar, es kann also keine Fehlerbehandlung für diesen Fall stattfinden.
- **Heruntergefallener Block**: Ein Block fällt während des Transports herunter.
  - Mögliche Maßnahme: Es sollte versucht werden, mit dem nächsten Block fortzufahren, anstatt zu versuchen, den verlorenen Block wiederzufinden.
- **Kollision mit Hindernissen**: Der Roboter kollidiert mit einem Hindernis.
  - Mögliche Maßnahme: Der Roboter fährt aus der Kollisionsrichtung aus.
  - Mögliche Maßnahme: Der Roboter führt einen Notstopp durch.
- **Instabilität des Stapels**: Es wird erkannt, dass ein Stapel instabil wird.
  - Mögliche Maßnahme: Der oberste Lego-Block wird neu positioniert.
  - Mögliche Maßnahme: Es wird ein neuer Stapel begonnen.

Detaillierte Anforderungen zur Fehlerbehandlung sind innerhalb des ***Projektteam***s abzustimmen und sollten in das Pflichtenheft aufgenommen werden.

7 Qualitätsanforderungen

Die Qualitätsanforderungen definieren die nicht-funktionalen Aspekte des Systems, die für einen erfolgreichen Betrieb erforderlich sind. Allgemein kann angenommen werden, dass Prioritäten `Genauigkeit = Zuverlässigkeit > Geschwindigkeit > *Ressourcen*effizienz` gelten.

Zusätzlich hat das ***Projektteam*** im Vorsemester bereits eigene Qualitätsmerkmale definiert. Diese müssen im Rahmen dieses Projekts überprüft werden. Ausmaß und Priorität wurden zum aktuellen Zeitpunkt noch nicht festgelegt.

7.1 Genauigkeit

Die Genauigkeit beim Stapeln der Lego-Blöcke muss ausreichend sein, damit die Stapel nicht umfallen und stabil stehen bleiben, auch nachdem der Roboter seine Aufgabe beendet hat. Dies erfordert eine präzise Steuerung der Bewegungen des Roboters und eine genaue Positionierung der Blöcke beim Stapeln.

7.2 Geschwindigkeit

Der Roboter bekommt einen fix definierten Zeitslot beim Wettbewerb zur Verfügung. Dieser Zeitslot darf nicht überschritten werden und wird später genauer bekanntgegeben.

7.3 Zuverlässigkeit

Der Roboter darf eine Fehlerrate von 100% haben, anzustreben ist eine Fehlerrate von 0%. Die akzeptable Fehlertoleranz wird durch den Wettbewerb, also den Vergleich zu anderen Projektgruppen, entschieden. Die absolute Fehlerrate hat keinen Einfluss auf die Bewertung des Projekterfolgs.

Die Robustheit des Systems gegenüber leichten Variationen in der Umgebung oder den Eigenschaften der Blöcke ist ein wichtiger Aspekt der Zuverlässigkeit und sollte durch umfangreiche Tests sichergestellt werden.

7.4 Sicherheit

Es sind keine Sicherheitsfunktionen wie etwa ein Notstopp-Knopf oder eine Pause-Funktion erforderlich.

Es muss jedoch verhindert werden, dass der Roboter selbst von der Arbeitsfläche fällt.

7.5 *Ressourcen*effizienz

Es dürfen so viele *Ressourcen* wie nötig für einen Durchlauf verwendet werden, solange der Roboter in der Lage bleibt, seinen Durchlauf zu beenden.

8 Anforderungen an die Projektentwicklung

Dieses Kapitel beschreibt die methodischen und organisatorischen Anforderungen an die Durchführung des Projekts.

8.1 Entwicklungsmethodik

Die Entwicklung orientiert sich an einem klassischen (nicht agilen) Vorgehensmodell. Im Rahmen des Projekts sind ein Lastenheft sowie ein Pflichtenheft zu erstellen. Zusätzlich müssen ein Organigramm sowie Stellenbeschreibungen des ***Projektteam***s erarbeitet werden, um die Verantwortlichkeiten und Kommunikationswege klar zu definieren.

8.2 Dokumentation

Neben den im obigen Abschnitt 8.1 angeführten Dokumenten müssen einzelne Dokumente im Rahmen der Lehrveranstaltung erstellt werden. Die konkreten Anforderungen an die Dokumente werden dabei näher definiert. Zusätzliche Dokumentation erfolgt im Eigeninteresse des ***Projektteam***s, um ein möglichst reibungsloses Projektmanagement zu gewährleisten.

Informationen, die erst zu einem späteren Zeitpunkt bekannt gegeben werden, sollten im Pflichtenheft aufgenommen werden, können aber auch separat festgehalten werden.

Alle Dokumente, die im Rahmen dieses Projekts erstellt werden, müssen sich an folgende Richtlinien halten:

- Konsistentes Layout
- Befolgen des Namensschemas
- Korrekte Versionierung `x.y`
  - x Inkrement für offiziell publizierte oder abgegebene Versionen
  - y Inkrement für im ***Projektteam*** intern publizierte Versionen
- Deckblatt bei mehr als 3 Seiten
- Korrekturlesung durch ein anderes Teammitglied

8.3 Tests

Die Tests werden vom ***Projektteam*** manuell in einer selbst nachgestellten Umgebung durchgeführt. Die Exzessivität der Tests sollte vom ***Projektteam*** selbstständig so gewählt werden, dass eine qualitative Umsetzung des Projekts sichergestellt werden kann.

8.4 Kommunikationswege

Es gibt bevorzugte Kommunikationswege für die verschiedenen Stakeholder.

8.4.1 Kursleiter

Kommunikationsweg	Zeitlicher Rahmen	Unter welchen Umständen
Mündlich	Unmittelbar zu Vorlesungen oder Übungen	Bei schnellen Fragen oder Anliegen
E-Mail	• Rund um die Uhr • Antwortet meist innerhalb von 24h	• Bei weniger dringenden Anliegen • Bei umfangreicheren Themen
Koordinationsgruppe	Hauptsächlich zu geplanten Besprechungen	Bei Anliegen (fast) aller Projektgruppen betreffend

Tabelle 3: Kommunikationswege mit **Kursleiter**

Der **Kursleiter** ist grundsätzlich mit der Festlegung zusätzlicher Kommunikationswege einverstanden. Dies erfolgt auf Absprache und wird von der Projektgruppe **G12** initiiert.

Der **Kursleiter** versucht, typisches Kommunikationsverhalten von realen Auftraggebern zu imitieren. Das beinhaltet:

- Manchmal zu beschäftigt für eine Antwort
- Schlechtere Antworten bei unpräzisen Fragen
- …

8.4.2 Partnergruppe G11

Grundsätzlich ist die Kommunikation mit der **Partnergruppe** G11 der behandelnden Gruppe **G12** überlassen. Es muss sich hierbei selbst koordiniert werden.

8.4.3 Gruppeninterne Kommunikation

Die gruppeninterne Kommunikation unterliegt dem gruppeneigenen Ermessen. Es gibt hierbei kein Standardprozedere oder andere Vorgaben.

9 Abnahmekriterien

Diese Projektausführung muss seine Anforderungen besser im Vergleich zu anderen Projektgruppen erfüllen. Es werden dafür die nachfolgenden Kriterien verwendet, wobei sich diese im Laufe der Zeit noch erweitern und konkretisieren können. Sie werden durch die **Koordinationsgruppe** und den **Kursleiter** festgelegt.

9.1 Dokumente

Geforderte Dokumente sind:

- Lastenheft
- Pflichtenheft
- *Organigramm*
- *Stellenbeschreibungen*

Zusätzlich gibt es individuell geforderte Dokumente.

Prinzipiell müssen alle Dokumente die im Abschnitt 8.2 angegebenen Kriterien erfüllen.

9.2 *Wettbewerb* bzw. Testtage

Genaue Kriterien werden erst am Tag des Wettbewerbs und am Alpha-Testtag bzw. Beta-Testtag entsprechend bekannt gegeben. Grob gestaltet sich die Bewertung wie folgt, wobei erstere Aspekte stärker gewichtet werden:

1. Erreichte Punkte steigen exponentiell mit der Höhe der Stapel
2. Die Einhaltung des zugeordneten Zeitslots entscheidet bei Gleichstand

Punkte können beim Testen in der offiziellen Testumgebung unter Aufsicht von Tutoren an Testtagen erzielt werden. Das geschieht unter den für diesen Tag festgelegten Kriterien.

9.2.1 Punktevergabe

Die Punktevergabe erfolgt nach folgenden Kriterien:

- 1. Stein des Stapels: 1 Punkt
- 2 Steine übereinander: 5 Punkte
- jeder weitere Stein: +3 Punkte
- Falschfarbiger Stein im Stapel: -X Punkte
- Stein endet außerhalb des *Zielbereich*s: −2 Punkte
- Unplatzierte Steine noch immer im *Steinebereich*: +0 Punkte

Dabei wird der höchste Stapel gezählt.

9.2.2 *Alpha-Testtag*

Am Alpha-Testtag werden verschiedene Farben getestet, um zu prüfen, welche zuverlässig vom Roboter erkannt werden können. Das Ziel ist, dass mindestens drei Farben zuverlässig erkannt werden. Dabei werden auch Einflussfaktoren wie Lichtverhältnisse berücksichtigt und getestet.

Der Startbereich des Roboters wird noch festgelegt, wobei Wünsche des **Projektteam***s* berücksichtigt werden können. Alle Ergebnisse werden dokumentiert und gesammelt, um eine fundierte Basis für die Weiterentwicklung zu schaffen.

Es ist zu beachten, dass das Fahrzeug nicht zwingend Kurven fahren muss, sondern lediglich vorwärts und rückwärts fahren können sollte. Die Startposition des Roboters ist noch offen und kann zufällig,

von einer anderen Gruppe gewählt oder selbst bestimmt sein. Die endgültige Entscheidung wird noch festgelegt.

Für den Erfolg des Projekts ist das Testen essenziell – je mehr Tests durchgeführt werden, desto besser wird das Endergebnis sein.

10 Design

Dieses Kapitel beschreibt die konkret geplante Umsetzung des zu entwickelnden Lego-Mindstorms-Roboters, basierend auf der Aufgabenstellung und den gegebenen Rahmenbedingungen. Die Abschnitte unterteilen sich in Hardware, Programmierung sowie Ablaufsteuerung.

10.1 Hardware

10.1.1 Basisplattform

Der Roboter basiert auf einer feststehenden Plattform.

Bewegung erfolgt über einen parallel betriebenen Kettenantrieb entlang einer translativen Achse (vor/zurück).

Diese Konfiguration erlaubt das sequentielle Abarbeiten der Blöcke auf dem *Steinebereich*.

10.1.2 Greifarm

Zentral montiertes Drehgelenk erlaubt das Schwenken über mehr als 180°, wodurch der Arm von der Entnahme- zur Ablagestelle bewegt werden kann.

Der Arm ist mit einem Greifer ausgestattet, der über eine mechanische Klammer die Blöcke aufnimmt.

Der Farbsensor ist direkt am Arm montiert, über oder unter dem Greifer, um die Farbe jedes Blocks vor dem Ablegen zu bestimmen.

10.1.3 Peripherie

- 3x Motoren:
  1. Antrieb der Plattform (vor/zurück)
  2. Drehgelenk des Arms
  3. Greifmechanik
- 1x Farbsensor für Farberkennung (RGB)
  - 2 verfügbare Farbsensoren:
    1. *NXT RGB-Sensor*
    2. *Neu RGB-Sensor* ← Gewählter Sensor

Es werden keine weiteren *Peripheriegeräte* benötigt. 4 Geräte belegen alle 4 Ports des EV3-Mono-Bricks, es können also keine weiteren Geräte angeschlossen werden.

10.1.3.a Vergleich der beiden Farbsensoren

Es wurden beide Farbsensoren getestet indem die Farbe der verschieden farbigen Blöcke mit beiden Farbsensoren ermittelt und dessen Zuverlässigkeit bewertet wurde.

Dabei konnte die Verwendung des *NXT RGB-Sensor*s ausgeschlossen werden, da dieser unter PyBricks nicht unterstützt wird.

Steinfarbe	NXT RGB	Neu RGB
Gelb	-	ok
Grün	-	ok
Rot	-	gut
Blau	-	ok
Weiß	-	nicht gewünscht

Skala:
1. gut
2. ok
3. nicht gewünscht

10.1.4 Skizze

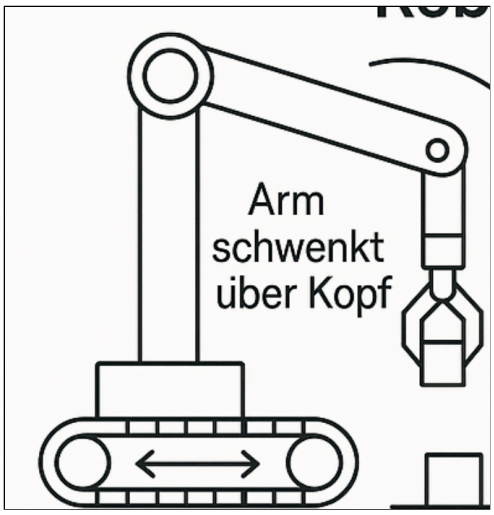

Abbildung 3: Roboter-Skizze

10.2 Programmierung

Programmiert wird in MicroPython unter Verwendung von PyBricks.

Aufbau nach modularer Struktur:

1. **Farberkennung**: Erkennung und Unterscheidung der Blockfarben
2. **Bewegungssteuerung**: Ansteuerung der Motoren für Vor-/Rückwärtsbewegung sowie Armsteuerung
3. **Greifmechanik**: Ansteuerung der Greifmechanik
4. **Positionsverwaltung**: Zuweisung von Ablagepositionen je Farbe, basierend auf ersten Fundstellen

10.3 Ablaufplan

10.3.1 Startkonfiguration

Der Roboter startet parallel zur Reihe der Blöcke im *Steinebereich*, mit idealem Abstand zur Greifposition.

Alle Steine können durch Translation entlang der Parallelen zum *Steinebereich* erreicht werden.

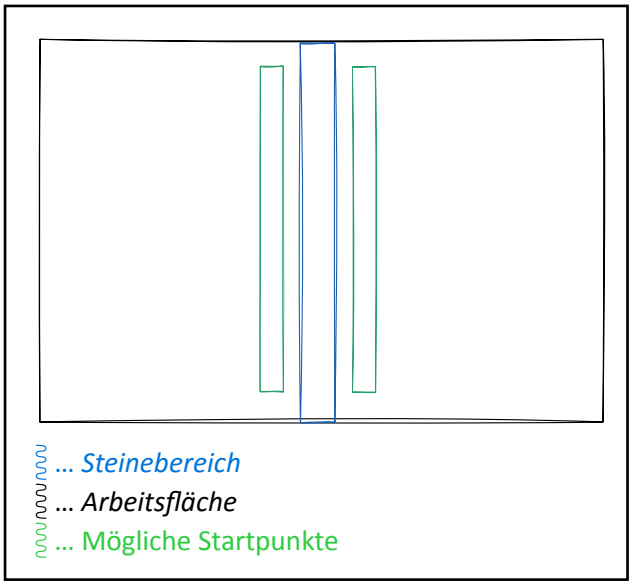

Abbildung 4: Zulässige Startpositionen des Roboters

10.3.2 Durchlauf

1. **Scan der Steine**: Der Roboter fährt entlang der Steine, bis ein Block erkannt wird.
2. **Greifen & Scannen**: Der Greifer nimmt den Block auf, der Farbsensor erkennt die Farbe.
3. **Zielposition ermitteln**:
   - Falls Farbe bereits begegnet: Verwende gespeicherte Zielposition.
   - Falls Farbe unbekannt: Lege neue Position fest und speichere sie.
4. **Transport & Ablegen**:
   - Roboter fährt zur Zielposition, Arm schwenkt über Kopf.
   - Block wird stabil abgelegt.
5. **Zurücksetzen & Wiederholen**: Arm schwenkt zurück, Roboter sucht nächsten Block.

10.3.3 Programmfluss

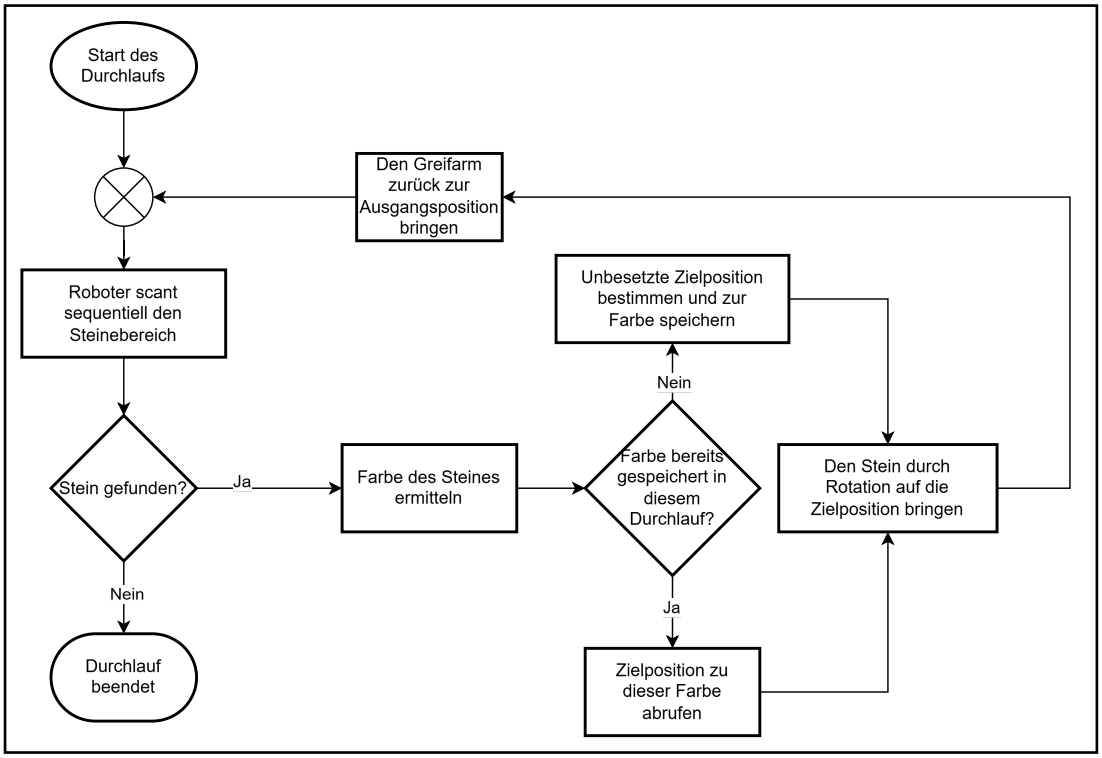

Abbildung 5: Programmfluss eines Durchlaufs

10.3.4 Teilabläufe

Detaillierte Beschreibung der einzelnen Teilabläufe aus dem Durchlauf.

10.3.4.a Steine zur Entnahme erkennen

Wie wird erkannt, dass es sich auf dem *Steinebereich* um einen Stein handelt, der gestapelt werden soll?

Der Farbsensor misst periodisch die Farbe nahe vor dem Sensor. Bei Erkennung einer Farbe, die einem potenziellen Stein entspricht, gilt ein Stein als gefunden. Da der Farbsensor auf dem Greifarm montiert ist, kann jener Stein gleich aufgehoben werden.

10.3.4.b Erkennung des Verlusts eines Steins aus dem Greifarm

Ist der Stein erfolgreich gehoben worden? Fiel der Stein unbeabsichtigt aus dem Greifarm?

Der Farbsensor wird verwendet, um periodisch die Farbe des Steins im Greifarm zu messen. Weicht diese vom Sollwert ab (Sollwert entspricht nach Aufheben der Farbe des aktuell zu transportierenden Steins), wissen wir, dass der Stein nicht erfolgreich gehoben wurde bzw. verloren ging.

Anhang

Teammitglieder der Gruppe G12

- Moritz Kieselbach
- Tim Wahlmüller
- Tim Peko
- Alexander Kranl
- Alexandra Usuanlele

Begriffe

Steinebereich	Fläche, auf der die Steine initial liegen; Fläche, von der die Steine entnommen werden
Zielbereich	Fläche, auf der die Steine nach Farben sortiert gestapelt werden
Arbeitsfläche	Fläche, auf der sich der Roboter befindet und er seine Aufgabe erfüllen muss; Voraussichtlich die gleiche Fläche wie der *Zielbereich*
Alpha-Version	Erste lauffähige Version des Roboters mit grundlegender Funktionalität
Alpha-Testtag	Tag des ersten Testens der *Alpha-Version* in der *Testumgebung*
Beta-Version	Funktionsfähige Version des Roboters, die noch Optimierungsbedarf bei nicht-funktionalen Anforderungen hat
Beta-Testtag	Tag des Testens der *Beta-Version* in der *Testumgebung*
Meilensteintag	Tag, an dem die Projektgruppen ihre aktuelle Lösung vorstellen und testen; Tage des *Wettbewerb*s, Alpha- und Beta-Testens
Wettbewerb	Tag des finalen *Wettbewerb*s; Alle Projektgruppen treten mit ihrer Lösung gegeneinander an
Pflichtenheft	Dokument, das die vom Auftragnehmer umzusetzenden Anforderungen detailliert beschreibt; es wird das konkrete "Wie?" beschrieben
Lastenheft	Dokument, das die Gesamtheit der Anforderungen des Auftraggebers an die Lieferungen und Leistungen eines Auftragnehmers enthält; es wird das "Was?" beschrieben
Ressourcen	Mittel, die zur Durchführung des Projekts benötigt werden, wie Hardware, Software, Zeit und Personal
Stakeholder	Personen oder Gruppen, die ein Interesse am Projektverlauf oder -ergebnis haben und diese direkt oder indirekt beeinflussen können
Projektteam	Die behandelnde Projektgruppe **G12**, die diese hiermit dokumentierte Projektausführung umsetzt
Betriebsumgebung	Die Umgebung, in der der Roboter seine Aufgabe erfüllen muss; Umgebung beim *Wettbewerb*
Testumgebung	Umgebung, in der die Funktionalität des Roboters getestet wird; Umgebung offiziell von Kursleitung bereitgestellt
Audimax	Hörsaal, in dem typischerweise die Vorlesungen und Übungen des Kurses stattfinden; Ort des finalen *Wettbewerb*s
Tutor	Höhersemestrige Studenten, die den **Kursleiter** im Rahmen der Lehrveranstaltung unterstützen
Peripheriegerät	Gerät, das an den EV3-Mono-Brick angeschlossen wird; Sensoren (z.B. Farbsensor, Gyrosensor) und auch Aktoren (z.B. Motoren)

NXT RGB-Sensor	Farbsensor, mit 3 Empfänger, beschriftet mit „RGB“; altes Modell; im Lego-Mindstorms-EV3-Set unter gleichen Namen enthalten
Neu RGB-Sensor	Farbsensor, mit 1 Empfänger; neues Modell; im Lego-Mindstorms-EV3-Set namenlos enthalten