



Softwareprojekt-Engineering, SS 2025

Protokoll

Design Review vom 30. April 2025

Gruppe 12

Autor

Tim Peko

Korrekturleser

Moritz Kieselbach

Teammitglieder

Moritz Kieselbach
Tim Wahlmüller
Tim Peko
Alexander Kranl
Alexandra Usuanlele

Inhaltsverzeichnis

Allgemeine Daten	3
Zeitpunkt	3
Teilnehmerliste	3
Ziel	3
Nächste Schritte	3
Besprechungsthemen	4
Meetingablauf	4
Roboter-Design	4
Programmierung	5
Ablaufplan	5
Anhang	7
Begriffe	7

Allgemeine Daten

Zeitpunkt

30. April 2025, 14:30 - 14:55

Teilnehmerliste

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

Ziel

1. ~~Korrektur des Lastenhefts gemeinsam durchgehen~~
2. Überzeugung des Arbeitgebers, vertreten durch Roxane, dass unsere Lösung funktioniert
3. Sanity-Check zur Machbarkeit unserer Lösung mit Roxane
4. Einholung von Feedback zu unserem Design durch Roxane

Nächste Schritte

Code Review: wann?

Tim Peko:

1. Korrektur vom Lastenheft einarbeiten
2. Überarbeitung des Protokolls
 - Nach Abschluss in Projektgruppe schicken für Feedback
 - Protokoll auf Moodle hochladen
3. Pflichtenheft fertigstellen

Besprechungsthemen

Meetingablauf

Wie läuft das Design Review ab?

1. Feedback zu diesem Protokoll
2. Überarbeitung dieses Protokolls
3. Abgabe des Protokolls auf Moodle

Feedback

- Inhalt ist gut
- Detaillierterer Ablaufplan
- Skizzen zur Veranschaulichung

Roboter-Design

Wie wird der Roboter physisch aufgebaut und zusammengesetzt sein? Welche Mechanik wird verwendet?

Der Roboter besteht aus einer festen Plattform, einem parallel betriebenen Kettenantrieb und einem Greifarm.

Mechanik und Aufbau

1. Basisplattform
 - Feststehende Plattform
 - Kettenantrieb zur Bewegung in einer translativen Achse
 - Ermöglicht es, den gesamten *Steinebereich* abzuarbeiten
2. Drehgelenk des Greifarms
 - Zentral montierter Motor ermöglicht Bewegung um rotatorische Achse
 - Erlaubt das Schwenken des Arms über den Roboter selbst ($>180^\circ$ Stellungsbereich)
 - Ermöglicht das Absetzen des Greifers
3. Armmechanik
 - Greifer: Wird mechanisch durch dritten Motor betrieben
 - Farbsensor: Direkt am Arm über/unter dem Greifer montiert; erkennt die Farbe des Steins vor dem Ablegen

Skizze

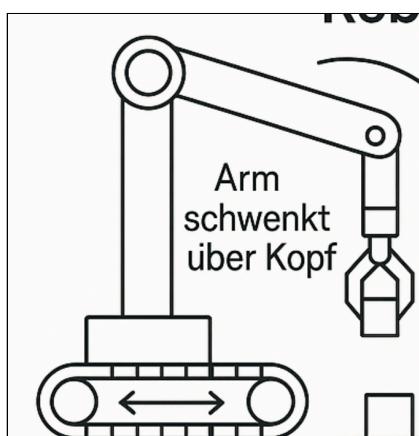


Abbildung 1: Roboter-Skizze

Verwendete Peripheriegeräte

- Farbsensor
- 3x Motoren
 - 1. Vor-/Zurückfahren des gesamten Roboters
 - 2. Auf- und Abbewegung des Arms
 - Über Kopf schwenken möglich
 - 3. Offnen und Schließen des Greifers

Programmierung

Wie wird der Roboter programmiert? Welche Programmiersprache wird verwendet? Welche Design-Prinzipien werden angewendet?

MicroPython auf PyBricks

Die Programmierung orientiert sich an modularen Prinzipien und ereignisgesteuerten Abläufen. Die Steuerlogik ist in mehrere Module wie

- Farberkennung
- Bewegungssteuerung
- Positionsverwaltung

unterteilt.

Ablaufplan

Wie wird der Roboter gesteuert? Welche Befehle werden verwendet? Wo befindet sich sein Startpunkt? Wie erreicht er sein Ziel?

Idee

Startpunkt: der Roboter startet im idealen Abstand zum *Steinebereich* und wird so positioniert, dass er durch bloßes Vor- und Zurückfahren alle Steine abarbeiten kann. → Parallel zur gebildeten Steinreihe auf dem *Steinebereich*.

1. **Ermittlung eines validen Steins aus dem *Steinebereich*:** der Roboter scannt den *Steinebereich* sequentiell bis er einen Stein erkennt. Dieser wird dann aufgehoben.
2. **Ermittlung der Farbe des Steins:** der Roboter scannt die Oberfläche des Steins mit dem Farbsensor, der ebenfalls am Greifarm montiert ist.
3. **Ermittlung der Zielposition:** Wenn der Roboter diese Farbe bereits einmal abgearbeitet hat, wird die dazu abgespeicherte Position verwendet. Andernfalls wird eine neue Zielposition ermittelt und zur jeweiligen Farbe abgespeichert.
4. **Ablage im *Zielbereich*:** Der Roboter schwenkt den Arm über Kopf und positioniert sich durch Vor-/Zurückfahren an die Zielposition im *Zielbereich*. Zum Schluss wird der Stein abgelegt.
5. **Wiederholung:** Der Roboter schwenkt den Arm zurück und wiederholt den gesamten Vorgang bis keine Steine mehr im *Steinebereich* erkannt werden.

Flowchart

Dieses Flowchart zeigt den logischen Ablauf eines Durchlaufs:

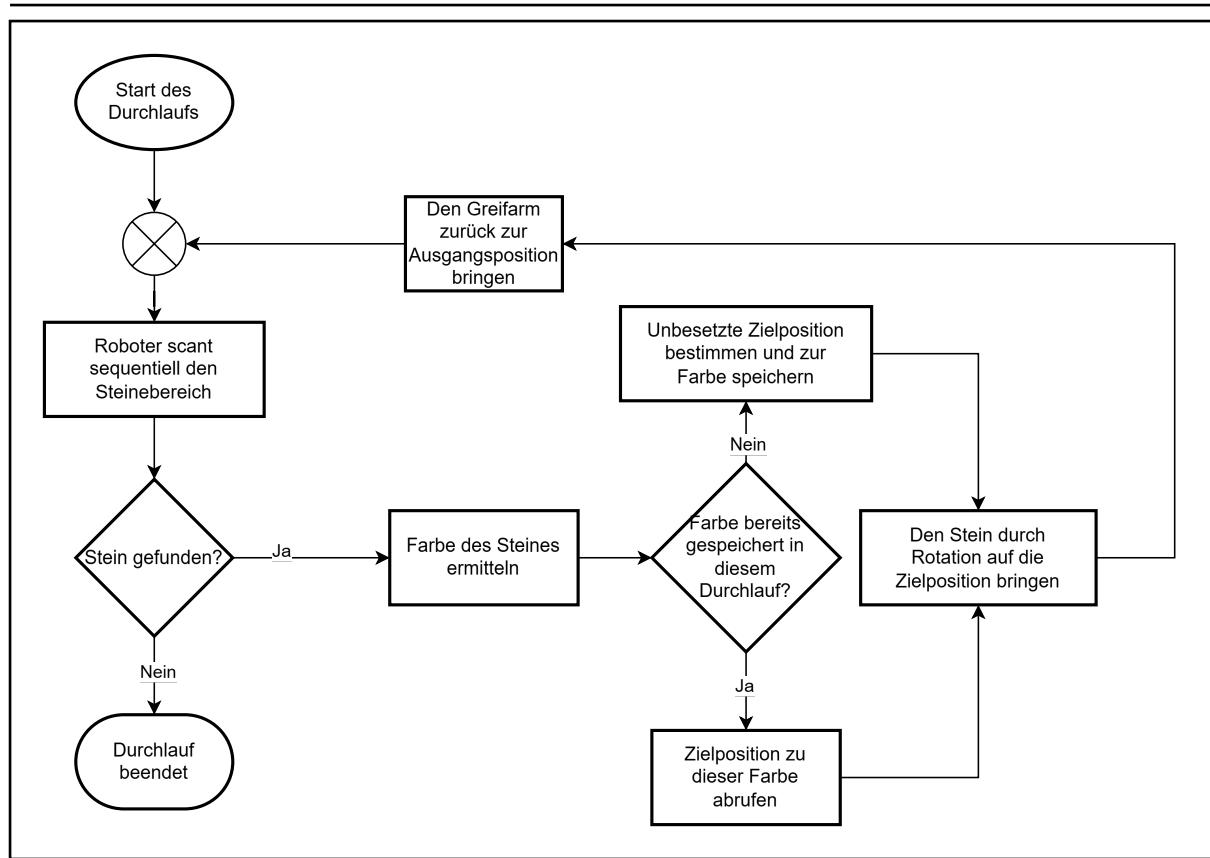


Abbildung 2: Programmfluss eines Durchlaufs

Anhang

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; Gleiche Fläche wie der <i>Zielbereich</i>