

Faculty of Computer Science, Institute of Systems Architecture, Chair of Systems Engineering

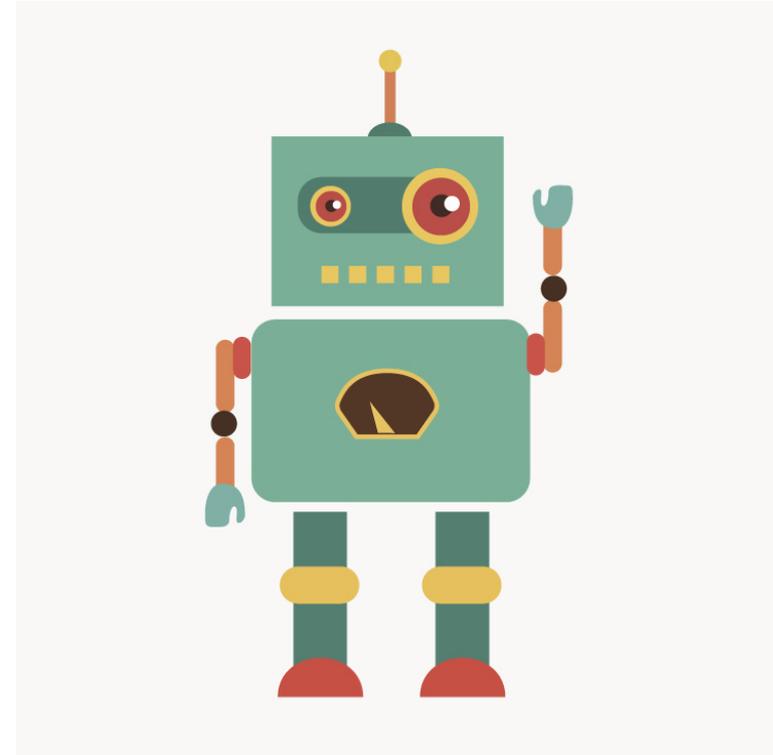
RoboLab Spring Course

Einführungsveranstaltung

von M.Sc. Samuel Knobloch

Gliederung

- Überblick, Inhalte und Ziele
- Aufgabenstellung
- Dokumentation
- Bewertungskriterien und Prüfung
- Organisatorisches
- Hinweise und Tipps



Überblick, Inhalte und Ziele

Überblick: Was ist das RoboLab?

- Interdisziplinäres Vollzeit-Praktikum mit Inhalten aus verschiedenen Fachbereichen
 - Robotik, Sensorik
 - Mathematik / Algorithmen
 - Softwareentwicklung / Projektplanung
 - Agile Methoden und Dokumentation
- Möglichkeit, eigene Schwerpunkte für das weitere Studium herauszufinden

Inhalte: Allgemein

- LEGO® Mindstorms EV3 Roboter-Programmierung
- Line-Following und Shortest-Path – Algorithmen
- Client-Server Kommunikation über MQTT
- Intelligente Erkundungsstrategien
- Agiles Arbeiten und Projektmanagement

Inhalte: Konkreter

- Programmierung eines LEGO® EV3 „Brick“ mit angepasstem Linux *ev3dev-robotlab*
- Teamarbeit in 3er-Gruppen
- Umsetzung der Aufgabenstellungen in Python3
 - Erkennen und Verfolgen von Linien und Knoten
 - Kommunikation mit Server über Netzwerk (WLAN)
 - Odometrie und Kartographie
 - Hinderniserkennung
 - Finden kürzester Wege
- Versionsverwaltung und Unit-Tests

**Wichtig: Ihr braucht
einen Laptop oder
PC!**

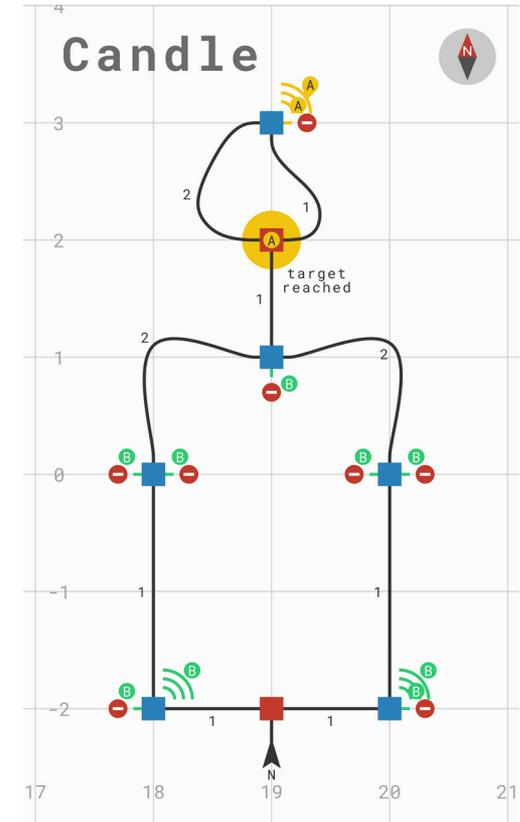
Ziele des Praktikums

- Sammeln von Erfahrungen in:
 - Anforderungsanalyse und Erarbeitung von Lösungsstrategien
 - Softwareentwicklung im Team
 - Programmierung und Nutzung unterschiedlicher Hilfsmittel
- Erweiterung der Programmierkenntnisse
- Heranführen an Sensorik, Robotik und verschiedene Problemstellungen der Informatik

Aufgabenstellung

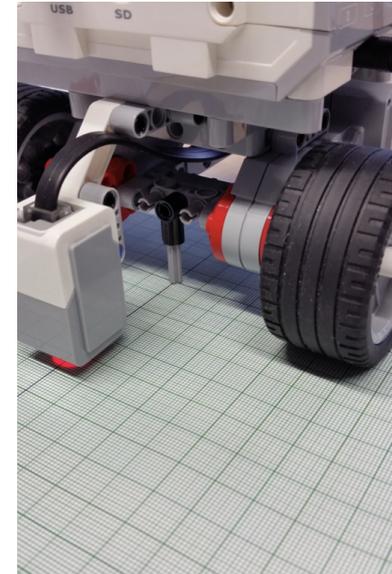
Aufgabenstellung: Eure Mission

- Roboter werden auf einem Planeten abgesetzt und sollen diesen erkunden
 - Linien müssen verfolgt, Hindernisse erkannt und Knoten verarbeitet werden
- Kartographieren des Planeten
 - Nutzen von Odometrie zur Positionsermittlung
 - Berechnung der kürzesten Wege zwischen entfernten Knoten
- Kommunikation mit dem „Mothership“ zum Abgleich der Daten an Knoten
 - Ermittelte Position senden, ggf. Korrektur erhalten und übernehmen
 - Pfadgewicht übernehmen



Aufgabenstellung: Fahrmechanik und Odometrie

- Linien- und Knotenerkennung mittels Farbsensor
- Schnelles und zuverlässiges Fahren auf Geraden und Kurven
- Richtungsbestimmung an Knoten
- Ermitteln der Position und Orientierung auf Planeten
 - über Winkelgeschwindigkeiten und Radumfänge
 - **Einfach:** Die grundlegende Implementierung
 - **Schwer:** Perfektionieren der Berechnung mit nur minimalen Abweichungen



Sensoren sind ungenau

Aufgabenstellung: Kommunikation

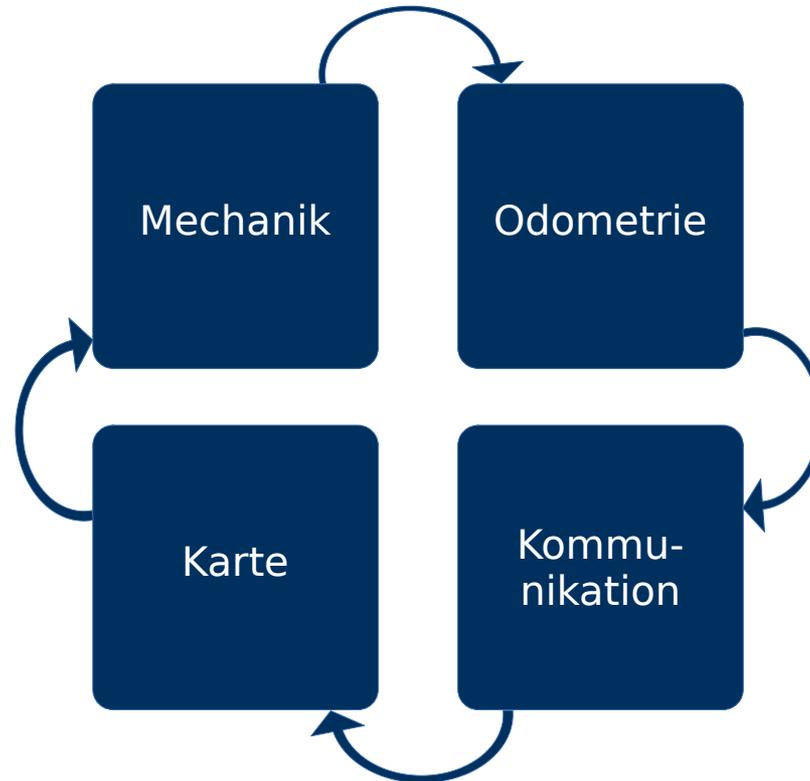
- Implementieren der Kommunikation mit einem PubSub-Server via MQTT
- Dabei sind ein paar Kleinigkeiten zu beachten:
 - Kommuniziert, also Nachrichten verarbeitet, wird nur auf Knoten
 - Sendet die Nachrichten auf den korrekten Kanälen
 - Achtet auf Einhaltung der JSON-Syntax
- **Tipp:** Es reicht, die *Verbindung einmalig* zu erstellen und erst *am Ende* wieder zu schließen
- Beachtet, dass es einen **Test-** und einen **Prüfungsmodus** gibt
 - Keine *debug*-Nachrichten in der Prüfung
 - Planetennamen nicht fest einprogrammieren

Aufgabenstellung: Template und Unit-Tests

- Wir stellen euch ein Projekt mit Klassen-Templates bereit
 - Vordefinierte Klassen und Methoden-Prototypen sind zu nutzen (*prüfungsrelevant*)
 - Die Signaturen dürfen nicht verändert werden
- Macht euch mit dem Deploy-Skript vertraut und richtet es korrekt ein
- Testet regelmäßig auf den Planeten
- Testet euren Quellcode frühzeitig
 - Befüllt die vorhandenen Planet-Tests (*prüfungsrelevant*)
 - Unit-Tests helfen, Code-Teile unabhängig auf korrekte Funktion zu testen

Alle Details findet ihr in der Dokumentation

Aufgabenstellung: Zusammenspiel der Komponenten



RoboLab-Dokumentation

Ein paar Worte zur Dokumentation

- Erreichbar unter <https://robolab.inf.tu-dresden.de>
- Enthält alle Informationen zu
 - detaillierter Aufgabenbeschreibung
 - Sensoren, Motoren, Brick
 - Betriebssystem, Template und Deploy-Skript
 - Prüfungshinweise und Bewertungskriterien
 - uvm.

Lest die Dokumentation!

Erste Schritte

- Macht euch vertraut mit:
 - der RoboLab-Dokumentation
 - den zu nutzenden Tools (IDE, Shell, ...)
 - Sensor- und Motordaten von LEGO®
- **Lest die Dokumentation** und erarbeitet euch die Teilaufgaben, die es zu lösen gilt
 - Link: <https://robolab.inf.tu-dresden.de/spring/>
- Richtet euren GitLab-Account auf unserem Server ein
 - Wird automatisch erstellt, wenn ihr unseren Test löst
- Konfiguriert euer System korrekt, um arbeiten zu können



Bewertungskriterien und Abschlussdemonstration

Bewertungskriterien

- Kriterien sind ausführlicher in der Dokumentation gelistet
 - Link: <https://robolab.inf.tu-dresden.de/spring/exam/criteria>
- Vier Bereiche sind maßgeblich:
 - **Fortbewegung** – Kann der Roboter verschiedenen Streckentypen folgen und von Station zu Station fahren?
 - **Kommunikation** – Werden alle Nachrichtentypen und Kanäle korrekt verarbeitet?
 - **Karte** – Passen Datenstruktur, kürzeste Wege und ist die Erkundung „intelligent“?
 - **Meta** – Unit-Tests, Teamarbeit, Logbuch, persönliche Lernkurve
- Bestehensrelevant sind weiterhin einige zu erreichende Milestones
- Alle Kriterien sind nach Bewertung zur Einsicht einsehbar

Bewertungskriterien: Milestones

- 0) Teilnahme an Einführungsveranstaltung und Gruppenzuteilung [**P**]
- 1) Vorstellung der Aufgabenverteilung und Analyse einer Planetenkarte
- 2) Zwischenstandspräsentation erster Dienstag
 - jeweilige Aufgabenteile wurden angefangen zu implementieren
 - für noch fehlende Teile sind Lösungsansätze grob vorhanden
 - nachvollziehbarer Gesamtplan für weiteres Vorankommen vorhanden (bei Allen)
- 3) Zwischenstandspräsentation zweiter Donnerstag
 - alle Aufgabenteile weitestgehend fertig implementiert
 - für fehlende Teile sind Lösungsansätze vorhanden und erklärbar
- 4) Eigenleistung im Gruppenkontext [**P**]
- 5) Teilnahme an täglichen Stand-ups [**P**]
 - min. 75% Anwesenheit für Bestehen erforderlich

Abschlussdemonstration

- Roboter-Demonstration ist Gruppenleistung, Benotung am Ende jedoch Einzelleistung
- Zur Demonstration des Roboters am Abschlussstag besteht für alle Gruppenmitglieder Anwesenheitspflicht
- Wie wird geprüft:
 - Sperren der Repositories 8:00 am Abschlussstag
 - Prüfung wird mit dem aktuellen Stand im *master*-Branch durchgeführt
 - Mittlere Planetenkarte mit mehreren unterschiedlichen Szenarien
 - Testet Funktionen des Roboters und Verhalten in unvorhersehbaren Situationen
 - Ausführen der Unit-Tests (eure und unsere) im Nachhinein
- Der Schwerpunkt liegt auf den Szenarien auf dem Mittleren Planeten
 - Bringt ca. 50% der Gesamtpunkte, Rest verteilt sich auf Odometrie-Strecke, Unit-Tests und Meta

Organisatorisches

Organisatorisches: Weiterer Ablauf

- ~10:20 Uhr – Selbsteinschätzung mit kleinem Test im Anschluss (*Teilnahme prüfungsrelevant*)
 - Bearbeitungszeit: 10-15 Minuten
- ~13:00 Uhr – Freigabe der Gruppenaufteilung
- ~13:15 Uhr – Treffen der Gruppe im APB in den Räumen
 - Erste Absprachen innerhalb der Gruppe
 - Einlesen in die Dokumentation, GitLab-Aufbau usw.
 - Einrichten der eigenen Arbeitsumgebung (PyCharm, WSL2 etc.)

Organisatorisches: Tagesziele

- Lernt euch in euren Gruppen kennen
 - Wer hat den Hut auf? Wie organisiert ihr euch?
- Lest die Dokumentation, verschafft euch Überblick und arbeitet „Erste Schritte“ ab
- Lest die Aufgabenstellung und macht euch erste Notizen

Ziel: Schafft die Grundlagen, damit es ab morgen richtig los gehen kann

Organisatorisches: Die nächsten zwei Wochen

- Tägliche Besprechung eures Standes mit den Tutoren (Stand-up Meeting) [**Milestone**]
 - Mit Tutoren vereinbarte Uhrzeit ist verbindlich
 - Notiert euer Vorankommen im Wiki (*bewertungsrelevant*)
- Sprecht Probleme und Schwierigkeiten (auch teaminterne) frühzeitig an
 - Nur so können wir euch helfen
- Wir sind Mo-Fr mindestens von 9:00 - 18:00 für euch da (11:30 – 13:00 ist Pause für uns)
 - Am Samstag bieten wir Betreuung nur nach Absprache an, *ist trotzdem Arbeitstag*
 - An Sonntagen bieten wir KEINE BETREUUNG an (Ausnahmen bestätigen die Regel)

Organisatorisches: Die nächsten zwei Wochen

- In erster Woche (bis Dienstag) besteht Möglichkeit zur fristgerechten Abmeldung
 - Bspw. wenn man eigene, zu große Wissenslücken hat
 - Bspw. wenn man doch parallel arbeiten muss und keine Zeit für das RoboLab findet
- Ab zweiter Woche (Mittwoch) ist alles Prüfungsleistung [**Milestones**]
 - Bedeutet: Anwesenheitspflicht (Absprachen mit Verantwortlichem sind möglich)!
 - Zwischenpräsentationen des Fortschritts

Organisatorisches: Abschlussdemonstration

- **Datum: 01.04.2025 ab 9:00 Uhr**
 - Gestaffelt in zwei Blocks: **9:00 – 12:30** und **13:30 – 17:00**
 - Mehrere parallele Stationen
- Wir bieten euch am Abend davor erweiterten Support für den Feinschliff
 - Bis ca. 23:00

**Anwesenheitspflicht für alle
Gruppenmitglieder!**

Hinweise und Tipps

Hinweise und Tipps: Verhalten an den Planeten

Die Herstellung von Planeten ist sehr zeitaufwendig.
Geht daher bitte mit den Planetenkarten, die ausliegen (werden), sorgsam um.

Insbesondere gilt:

- **Nicht auf die Planeten treten!**
Verschmutzungen können auch eure Lichtsensoren verwirren.
- **Die Planeten sind keine Picknickdecken!**
Haltet Speisen und Getränke von den Planeten fern (min. 3 Meter).
- **Stühle und Tische bleiben in den Räumen!**
Jede Regel hat ihre Geschichte.
Im Foyer können die „Lümmel“ genutzt werden.

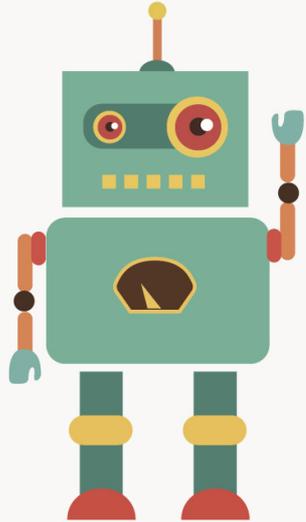
Bei Missachtung dieser Regeln müsst ihr mit ca. 8 Tutoren (mit bösen Blicken) um euch herum rechnen!

Hinweise und Tipps

- Comitted eure lokalen Stände regelmäßig und von Beginn an (*prüfungsrelevant*)
- Lest die Dokumentation. Mehrfach.
 - Wir freuen uns über Fehlermeldungen und Verbesserungsvorschläge
- Erstellt eine Liste der Anforderungen und Spezifikationen
- Plant euer Vorgehen:
 - Aufgabenteilung – Wer macht was?
 - Prüft euren Fortschritt regelmäßig (Unit-Tests, Planetentests)
- Geht iterativ vor
 - Aufgabenteile bedingen einander – versucht nicht, alles auf einmal zu lösen

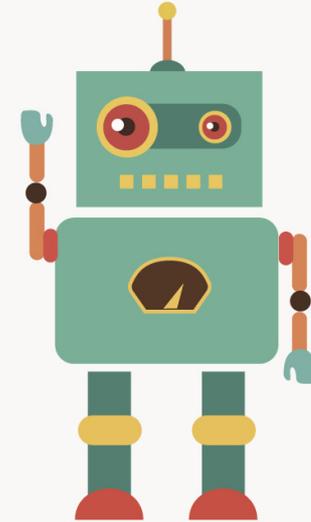
Hinweise und Tipps

- Codequalität ist wichtig (*bewertungsrelevant*)
 - Programmiert modular, objektorientiert und in Klassen
 - Schreibt sauberen, verständlichen Code
 - Habt Mut, kaputten Code zu löschen und „sauber“ neu zu implementieren
- Fehlertoleranz ist wichtiger als Geschwindigkeit
- Gute Teillösungen bringen euch mehr als eine mittelmäßige Gesamtlösung
- **Unterschätzt den Aufwand nicht – Das ist ein Vollzeit-Praktikum!**
- Redet miteinander und mit uns, aber kopiert / plagiiert nicht
 - Gilt auch für „KI“-Tools wie *ChatGPT*, *GitHub Copilot* o.ä. – **Nutzung gilt als Betrugsversuch (5.0)!**



Fragen?

Nein?



Viel Spaß beim RoboLab!

RoboLab Test and Group Tool

<https://robolab.inf.tu-dresden.de/groups>

