

Dokumentation

Roboter-Fangen

Maschinenbauinformatik 3. & 5. Semester

Michael Mertens, Jonah Vennemann, Sven Stegemann, Eugen Zwetzig

8. Februar 2016

Inhaltsverzeichnis

1	Projektbeschreibung	3
1.1	Spielablauf	4
2	Gantt-Diagramm	4
3	Aufwandsschätzung	7
4	GitHub	10
4.1	ZenHub	10
5	Bedienung	12
6	Programmierung	12
6.1	Programmablaufplan	12
6.2	Benutzeroberfläche	13
6.3	Klassen	13
6.3.1	mVektor	13
6.3.2	mTKI	13
6.3.3	mKonstanten	14
6.3.4	mRoboterDaten	15
7	Resümee	15

Abbildungsverzeichnis

1	Gantt-Diagramm	6
---	--------------------------	---

Quellcode

1 Projektbeschreibung

Bei dem Projekt „Roboter-Fangen“ für das Modul IT-Projektmanagement besteht unsere Aufgabe als eines von zwei Teams in der Programmierung einer Steuerungssoftware für das Fischertechnik ROBOTICS TXT Discovery Set.

Das Gemeinziel ist ein lauffähiges Fangen-Spiel zu erstellen bei dem vier Roboter pro Team von der jeweiligen Software gesteuert werden.

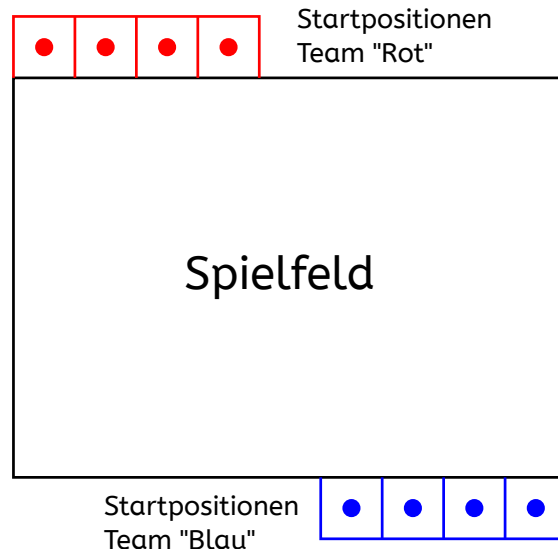
Dabei werden die Positionsdaten aller Roboter von einem Schiedsrichter-Server mit Hilfe einer Kamera berechnet und an die Steuerungssoftware der beiden Teams geschickt. Hauptbestandteile der Steuerungssoftware:

- Benutzeroberfläche:
 - Kamerabilder
 - Eingabefelder zum Verbinden
 - zusätzliche Informationen
- Positionsdatenverarbeitung über eine Vektorklasse:
 - Attribute: x,y als Typ Double
 - Methoden: Addieren, Subtrahieren, Skalar multiplizieren, Winkel berechnen
- Elemente der KI:
 - Fangen
 - Fliehen
 - Ausweichen
 - im Feld bleiben
 - Rausfahren nachdem Gefangenwerden

Neben der Programmierung gehören dabei auch die Planung, die Dokumentation des Codes sowie die Darstellung des Projekts dazu.

- Quelltextkommentare
- Präsentation
- Zeiterfassung
- Betriebsanleitung
- Spielregeln

1.1 Spielablauf



Es werden pro Gruppe 4 Roboter auf dem Spielfeld an ihre Startpositionen platziert. Die Größe des Spielfeldes ist festgelegt.

Alle Roboter, von beiden Teams, verbinden sich mit dem Server und übergeben diesem, dass Sie bereit sind. Sobald der Server das Startsignal gibt, fahren die Roboter mit einer Geschwindigkeit die $\frac{3}{4}$ der vollen Geschwindigkeit entspricht los.

Dann versuchen sich die Roboter, der beiden Teams, gegenseitig zu fangen. Befindet sich ein gegnerischer Roboter im Abstand X vor einem anderen Roboter, so wird die maximale Geschwindigkeit des hinteren Roboters freigeschaltet.

Wenn ein Roboter gefangen wurde, sendet dieser ein Signal an den Server. Dieser wiederum setzt den „gefangen“ Roboter auf nicht Aktiv. Ist ein Roboter nicht Aktiv, so fährt dieser aus dem Spielfeld und verbleibt dort eine gewisse Zeit, bis er in's Spiel zurück kehrt.

Folgende Regeln wurden getroffen:

- Ein Roboter darf erst losfahren, wenn der Server das Startsignal gegeben hat
- Ein Roboter darf sich nicht auf der Stelle drehen
- Ein Roboter darf nicht mit dem Rücken zur Wand stehen, da es so nicht möglich ist diesen zu fangen
- Erst ab einem Abstand X, darf die volle Geschwindigkeit zur Verfügung stehen

2 Gantt-Diagramm

Ein Gantt-Diagramm oder auch Balkenplan ist ein nach dem Unternehmensberater Henry L. Gantt benanntes Instrument des Projektmanagements, das die zeitliche Abfolge

von Aktivitäten grafisch in Form von Balken auf einer Zeitachse darstellt.

In der Abbildung 1 sieht man die einzelnen Aktivitäten, die wir für unser Projekt Roboter-Fangen eingeplant haben. Außerdem sieht

Einige Aktivitäten haben wir in Gruppen eingeteilt, um:

- Planung
- Programmierung - Teil 1
- Programmierung - Teil 2
- Dokumentation

		Name	Dauer	Start	Ende	Vorgänger	Ressourcen
1	✓	M0: Start	0 tage	23.12.15 08:00	23.12.15 08:00		
2	✓	Analyse	0,5 tage	23.12.15 08:00	23.12.15 13:00	1	Gruppe Blau;Gruppe ...
3	✓	Aufgabenaufteilung	0,5 tage	23.12.15 12:00	23.12.15 17:00	2	Gruppe Blau
4	✓	Planung	0,5 tage	28.12.15 08:00	28.12.15 13:00	3	
5	✓	Projektbeschreibung	0,5 tage	28.12.15 08:00	28.12.15 13:00		Eugen Zwetzig;Com...
6	✓	Spielregeln	0,5 tage	28.12.15 08:00	28.12.15 13:00	5AA	Jonah Vennemann;Eu...
7	✓	Skizze PAP KI	0,5 tage	28.12.15 08:00	28.12.15 13:00	6AA	Michael Mertens
8	📅	M1: Planung abgeschlossen	0 tage	04.01.16 08:00	04.01.16 08:00	4	
9		Programmierung - Teil I	9,5 tage	04.01.16 08:00	15.01.16 13:00	8	
10		GUI-Design	1 tag	04.01.16 08:00	04.01.16 17:00		Computer;Jonah Venn...
11		Programmstart/verbinden	1,5 tage	05.01.16 08:00	06.01.16 13:00	10	
12		Positionsdaten empfang...	1 tag	06.01.16 13:00	07.01.16 13:00	11	Sven Stegemann
13		Fahrtrichtungen ermitteln	2 tage	07.01.16 13:00	11.01.16 13:00	12	
14		Als "gefangen" melden	1 tag	04.01.16 08:00	04.01.16 17:00	10AA	
15		Simple KI	4 tage	11.01.16 13:00	15.01.16 13:00	13	
16		KI - Fliehen	2 tage	11.01.16 13:00	13.01.16 13:00		Jonah Vennemann
17		KI - Ausweichen	2 tage	11.01.16 13:00	13.01.16 13:00	16AA	Eugen Zwetzig
18		KI - Im Feld bleiben	2 tage	11.01.16 13:00	13.01.16 13:00	17AA	Jonah Vennemann
19		KI - Fangen	2 tage	11.01.16 13:00	13.01.16 13:00	18AA	Eugen Zwetzig
20		KI - Rausfahren nachde...	2 tage	13.01.16 13:00	15.01.16 13:00	19	Jonah Vennemann
21	📅	M2: Erste Implementierung	0 tage	15.01.16 13:00	15.01.16 13:00	9	
22		Vorabpräsentation erstell...	1 tag	15.01.16 13:00	18.01.16 13:00	21	
23	📅	M3: Kurzpräsentation	0 tage	18.01.16 13:00	18.01.16 13:00	22	
24		Programmierung - Teil II	3 tage	18.01.16 12:00	21.01.16 13:00	23	
25	✓	Log-Funktion	0,5 tage	18.01.16 12:00	18.01.16 17:00		Jonah Vennemann
26		Kamerabilder anzeigen	1 tag	18.01.16 13:00	19.01.16 13:00	27AA	Michael Mertens
27	✓	Klasse zur Vektorberech...	1 tag	18.01.16 12:00	19.01.16 13:00	25AA	Eugen Zwetzig
28	📅	Tests	2 tage	19.01.16 13:00	21.01.16 13:00	27	Computer;Gruppe Bla...
29	📅	M4: Programmieren abge...	0 tage	21.01.16 13:00	21.01.16 13:00	24	
30		Präsentation abschließen	1,5 tage	21.01.16 13:00	22.01.16 17:00	29	Jonah Vennemann
31	📅	M5: Endpräsentation	0 tage	22.01.16 17:00	22.01.16 17:00	30	
32	📅	Dokumentation	30 tage	05.01.16 08:00	15.02.16 17:00	9AA	
33	📅	Betriebsanleitung	30 tage	05.01.16 08:00	15.02.16 17:00		Eugen Zwetzig;Com...
34	📅	M6: Dokumentation abge...	0 tage	15.02.16 17:00	15.02.16 17:00	32	

Roboter fangen Roboter

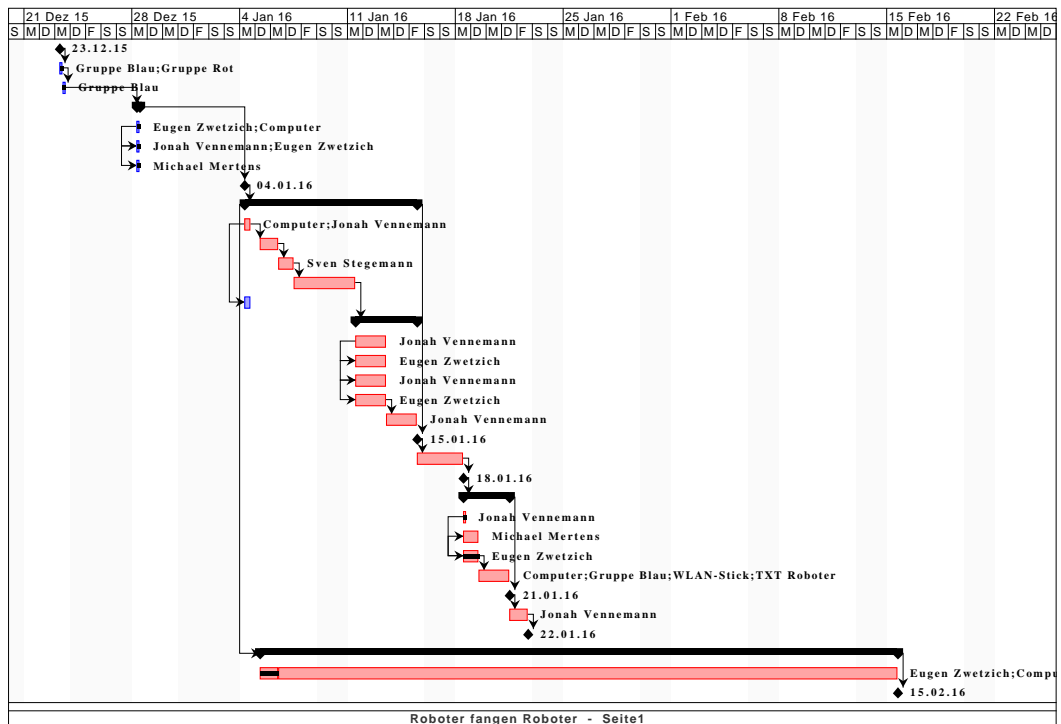


Abbildung 1: Gantt-Diagramm

3 Aufwandsschätzung

Um den Aufwand unseres IT-Projektes abschätzen zu können, haben wir die Methode Function-Point benutzt.

Das Function-Point-Verfahren(auch -Analyse oder -Methode, kurz: FPA) dient der Bewertung des fachlich-funktionalen Umfangs eines Informationstechnischen Systems.

Die Durchführung des Verfahrens verläuft in 5 Schritten:

1. Analyse der Komponenten und Kategorisierung ihrer Funktionalitäten
2. Bewertung der verschiedenen Funktionskategorien
3. Einbeziehung besonderer Einflussfaktoren
4. Ermittlung der sog. Total Function Points(TFP)
5. Ableitung des zu erwartenden Entwicklungsaufwandes

1. Schritt

- Eingabedaten
 - GUI
 - Programmstart
- Ausgabedaten
 - Ereignisprotokolldatei
 - Kamerabild
 - Steuerbefehle senden
- projektbez. Datenbestände
 - Fahrtrichtung
 - Fangen
 - Fliehen
 - Ausweichen
 - Im Feld bleiben
 - Rausfahren nach dem Fangen
 - Vektorberechnung
- externe Datenbestände

- Positionsdaten
- Mitteilung gefangen
- Roboter aktiv?

2. Schritt

Funktionskategorie	Anzahl der Funktionen			Faktoren der Funktionen			Funktionspunkte		
	Einfach	Mittel	Komplex	Einfach	Mittel	Komplex	Einfach	Mittel	Komplex
Eingabedaten	1	1	0	3	4	6	3	4	0
Ausgabedaten	1	2	0	4	5	7	4	10	0
Projektbez. Datenbestände	1	3	3	7	10	15	7	30	45
Externe Datenbestände	3	0	0	5	7	10	15	0	0

Summe S1:	118
------------------	-----

3. Schritt

Einflussfaktoren		Gewichte	
Nr			
1	Schwierigkeit und Komplexität der Rechenoperatoren (Faktor 2)	2	
2	Schwierigkeit und Komplexität der Ablauflogik	5	
3	Umfang der Ausnahmeregelung (Faktor 2)	6	
4	Verflechtungen mit anderen IT-Systemen	3	
5	dezentrale Verarbeitung und Datenhaltung	0	
6	erforderliche Maßnahmen der IT Sicherheit	0	
7	angestrebte Rechengeschwindigkeit	1	
8	Konvertierung der Datenbeständen	0	
9	Benutzer- und Änderungsfreundlichkeit	1	
10	Wiederverwendbarkeit von Komponenten (bspw. Klassen)	1	

Summe S2:	19
------------------	----

4. Schritt

$$\begin{aligned}\text{TFP}^1 &= S1 \cdot S3 \\ &= S1 \cdot \left(0,7 + \frac{S2}{100}\right) \\ &= 118 \cdot \left(0,7 + \frac{19}{100}\right) \\ \text{TFP} &= 105,02\end{aligned}$$

5. Schritt

$$\text{PM}^2 = 0,08 \cdot \text{TFP} - 7 \leq 1000 \text{TFP} > \text{PM} = 0,08 \cdot \text{TFP} - 108$$

$$\begin{aligned}\text{PM} &= 0,08 \cdot \text{TFP} - 7 \\ &= 0,08 \cdot 105,02 - 7 \\ \text{PM} &= 1,4016\end{aligned}$$

$$\text{PM} = 672,77\text{h}$$

$$3 \text{ Personen} = 224,256\text{h pro Person}$$

$$4 \text{ Personen} = 168,192\text{h pro Person}$$

⇒ Bei einem 4 starken Team benötigen wir ca. 170h pro Person.

4 GitHub

GitHub ist ein webbasierter Online-Dienst, für die Versionsverwaltungssoftware Git.

Git wurde von Linus Torvalds ursprünglich für die Verwaltung des Linux-Kernels geschrieben.

4.1 ZenHub

ZenHub ist ein Projektmanagement Addon für GitHub.

In diesem ist es möglich

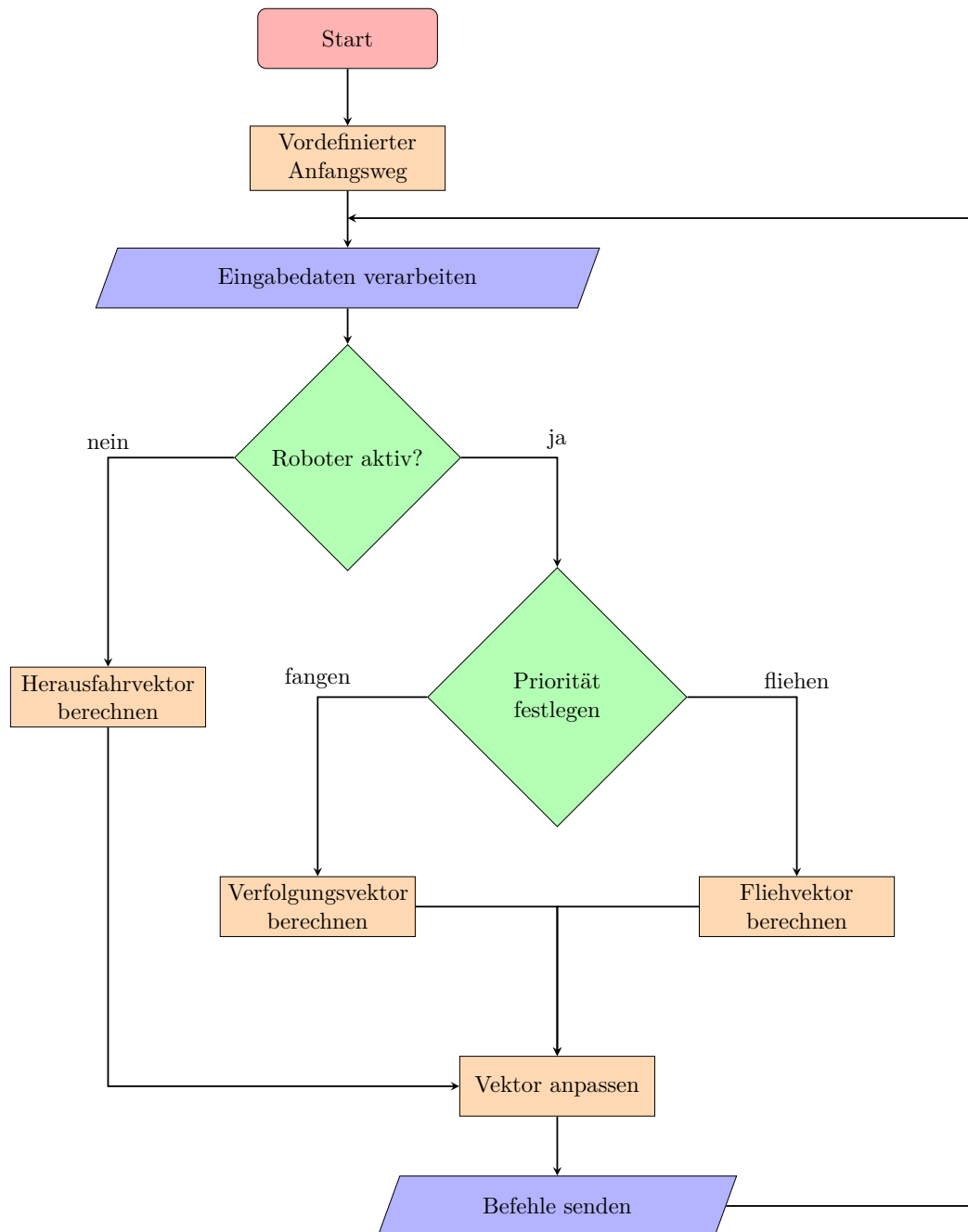
¹TFP=Total Function Points

²Personenmonate(PM) = 20 Arbeitstage

5 Bedienung

6 Programmierung

6.1 Programmablaufplan



6.2 Benutzeroberfläche

6.3 Klassen

Für die Berechnungen und Logik haben wir eigene Klassen geschrieben.

Diese unterteilen sich in:

- mVektor
- mTKI
- mKonstanten
- mRoboterDaten

6.3.1 mVektor

Die Klasse mVektor besteht aus dem Record TVektor.

Dieser hat folgende Funktionen, überladende Operatoren und Variablen:

- Funktionen
 - Winkel(überladen)** gibt den Winkel des Vektors, bezogen auf die X-Achse, zurück
 - Winkel(überladen)** gibt den Winkel zwischen zwei Vektoren zurück
 - Betrag** gibt die Länge des Vektors zurück(Euklidische Norm „2-Norm“)
 - Drehen** dreht den Vektor um einen, als Parameter übergebenen, Winkel
- Operatoren
 - Add** Komponentenweise Addition zweier Vektoren
 - Substract** Komponentenweise Subtraktion zweier Vektoren
 - Multiply(überladen)** Komponentenweise Multiplikation eines Vektors mit einem Skalar
 - Multiply(überladen)** Komponentenweise Multiplikation eines Skalars mit einem Vektor
 - Equal** überprüft die Komponenten zweier Vektoren auf Gleichheit
- Variablen
 - x,y** Komponenten des Vektors

6.3.2 mTKI

Die Klasse mTKI hat einen Datentypen TAction mit den Werten Fliehen und Fangen und eine abgeleitete Klasse TKI von TObject. Die abgeleitete Klasse TKI besteht aus folgenden Funktionen, Prozeduren und Variablen:

- Funktionen

PrioritätFestlegen Anhand der Positionsdaten der gegnerischen Roboter wird überprüft, welcher sich am nächsten an unserem Roboter befindet. Anschließend wird über die Winkel Funktion von der Klasse mVektor ermittelt, ob sich dieser Roboter vor oder hinter unserem Roboter befindet. Danach wird die Priorität auf FANGEN bzw. FLIEHEN gesetzt.

FangvektorBerechnen description

FliehvektorBerechnen description

AusweichvektorBerechnen description

RausfahrvektorBerechnen Sobald ein Roboter aus unserem Team gefangen wurde, wird ein Vektor zum Raus fahren aus dem Spielfeld berechnet.

ServerdatenEmpfangen description

Anmelden description

- Prozeduren

SteuerbefehlSenden Beschreibung

GeschwindigkeitBerechnen description

Initialisierung description

Steuern description

- Variablen

ZeitletzterFrames Beschreibung

RoboterDaten description

Roboter description

Spielfeld description

Server description

6.3.3 mKonstanten

Da wir an verschiedenen Stellen die gleichen Werte benötigten, erstellen wir eine eigene Klasse für Konstanten.

- Variablen

Mindestabstand Beschreibung

Nullvektor ist ein konstanter Record

6.3.4 mRoboterDaten

Um den Zugriff auf die Daten eines Roboters zu vereinfachen, haben wir diese in einer eigenen Klasse mRoboterDaten untergebracht.

Diese besteht aus einem Record TRoboterDaten mit folgenden Variablen:

- Variablen

Position eines jeden Roboters vom Typ TVektor

Geschwindigkeit eines jeden Roboters vom Typ TVektor

Positionverlauf ist eine Warteschlange(TQueue) mit Positionen des Roboters vom Typ TVektor

Aktiv description

7 Resümee