Flee and Catch

Nachstellung von Schwarmverhalten mittels Kleinroboter

von Manuel Bothner & Simon Lang

Aufgabenstellung

Abbildung grundlegendes Schwarmverhaltens mittels Lego MINDSTORMS Roboter anhand natürlicher Vorbilder:

- Fische
- Ameisen
- Vögel

Anforderungsdefinitionen

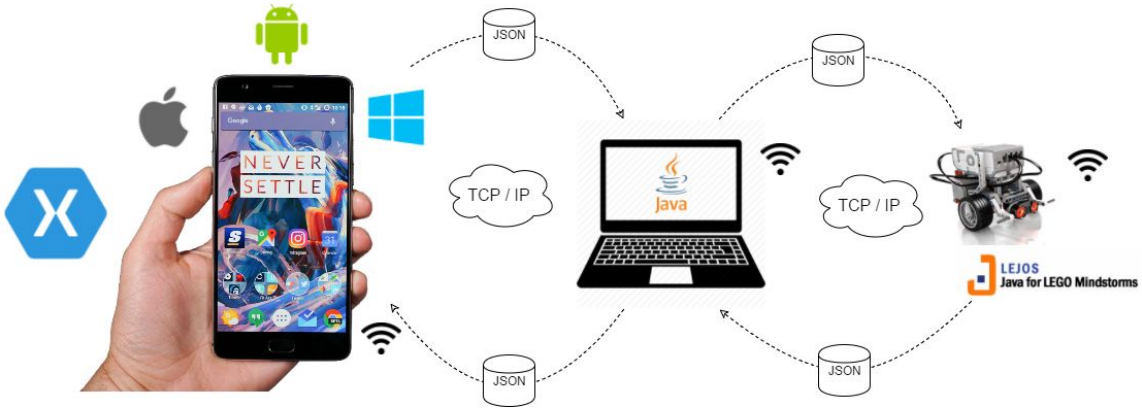
- Grundlegende Steuerungsfunktion
  - Kommunikationssystem
  - Datenbasiertes autonomes Verhalten
  - Roboterkinematik
- Abbildung von Schwarmverhalten
  - Synchrone Bewegung
  - Verfolgende Bewegung
- Nutzereinsicht
  - Szenariokonstellation
  - Erfassung der Roboterdaten

Softwarearchitektur

- Mobile Anwendung (App)
  - **Benutzerschnittstelle zur Steuerung und Überwachung**
  - Plattformübergreifende Entwicklung - Xamarin
  - Strukturierte Implementierung - MVVM
- Desktopanwendung (Backend)
  - **Basis des Kommunikationssystems**
  - Management der Geräte & Szenarien
  - Grafische Benutzerschnittstelle - JavaFX
- LEGO Mindstorm EV3 (Roboter)
  - **Realisierung des Schwarms**
  - Grundlegende Steuerungsfunktionen - leJos
  - Datenbasiertes autonomes Verhalten

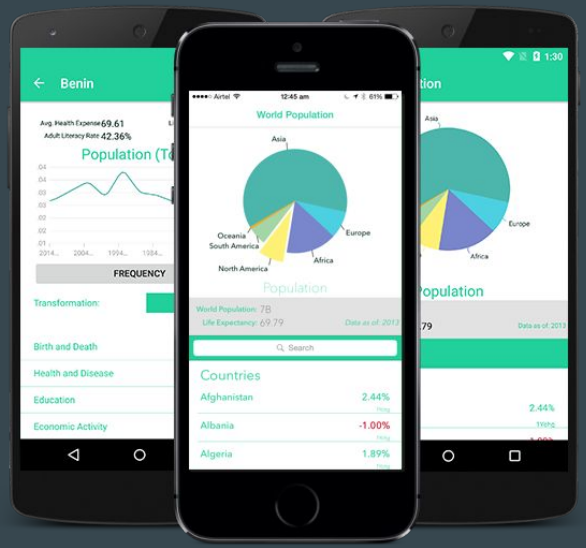
Grundlagen - Kommunikation

- Client - Server - Prinzip
  - Authentifizierung
- Datenbasierte Kommunikation - JSON
  - Definierte Kommandostruktur
  - Objektdarstellung
- Wireless mittels WLAN TCP / IP



Grundlagen - App

- Benutzerschnittstelle
  - Erstellung von Szenarien
  - Steuerung des Schwarms
  - Abrufen aktueller Daten
- Plattformübergreifende Implementierung
  - Graphical User Interface (GUI) - Xamarin Forms
  - Library - Portable Class Library (PCL)
- Model-View-ViewModel (MVVM)
  - Design Pattern zur strukturierten Implementierung
  - Trennung von View, Model und Business Logic
  - Verbindung der Daten (Binding) - Fody



Grundlagen - Backend

- Java-Anwendung (Konsole + JavaFX)
- Kommunikationszentrale
  - Erfassung / Verwaltung von Geräten
  - Szenarienverwaltung
- Zentrale Komponenten
  - Server
  - Interpreter
  - Controller-Klassen zur Verwaltung (Apps / Robots / Szenarios)
- GUI zur Dateneinsicht
  - Nachverfolgung & Überwachung
  - Debugging

Grundlagen - Roboter

LEJOS
Java for LEGO Mindstorms

- Multithreading für Kommunikation, Sensordatenerfassung, Steuerung
- Zentrale Komponenten Client & Robot-Controller
  - Client für Kommunikation
    - Empfangen von Steuerbefehlen
    - Empfangen von Daten (Position / Orientierung / Geschwindigkeit)
    - Senden eigener Daten
  - Robot-Controller für Steuerung und Sensordatenerfassung
    - Umsetzung der Steuerbefehle
    - Autonome, datenbasierte Navigation
    - Kontinuierliche Erfassung von Roboterdaten
- Ansteuerung der Sensoren / Motoren via leJOS-Bibliothek

Schwarmszenarien

- Synchrone Bewegung
  - Z. B. V-Formation bei Vögeln
  - Realisierung durch zeitgleiche Steuerkommandos
- Verfolgung
  - Asynchrone datenbasierte Bewegung
    - Verfolgung des führendes Roboters
    - Autonomes Bewegungsabbildung
- Fangen
  - Spielerisches Schwarmverhalten
  - Erweiterung der Verfolgung
    - Positionsbasiertes Fangen

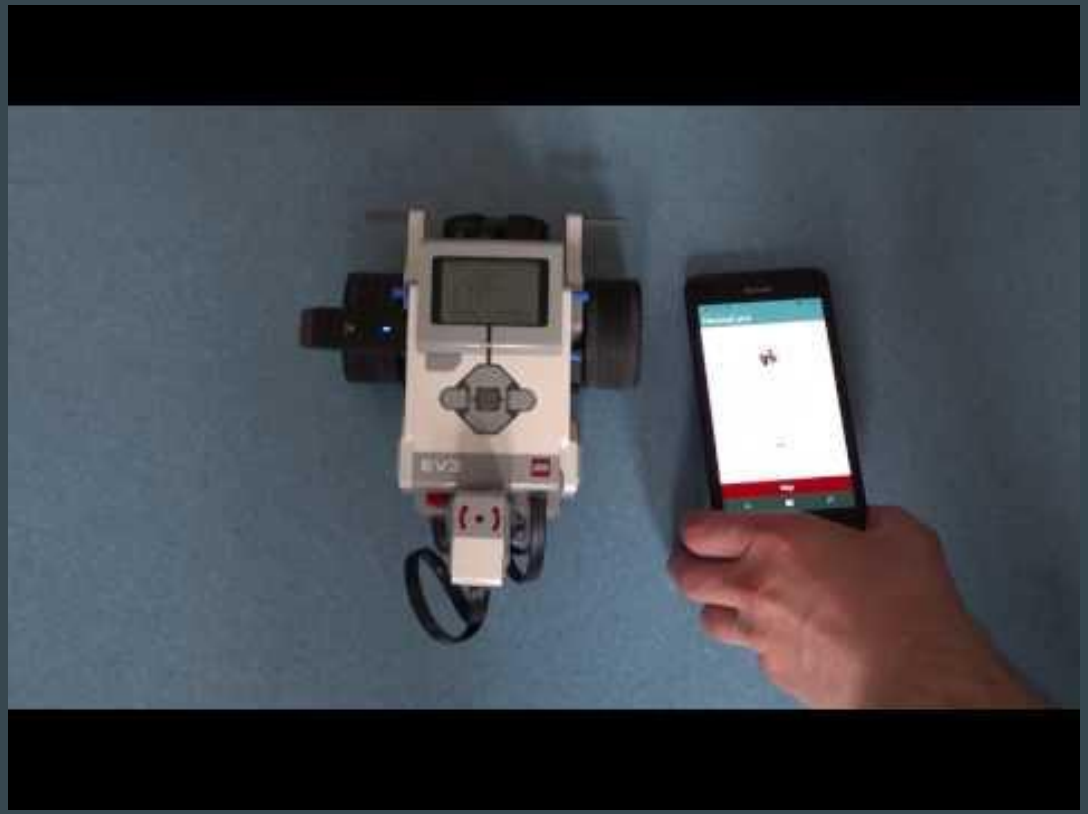
Grundlegender Szenarioablauf

- Backend Starten
  - Wartet auf die Anmeldung der Devices
- Initialisieren der Roboter
  - Starten (leJOS)
  - Auswählen des Robotertyps
  - Typabhängige Initialisierung & Anmeldung am Backend
- Kontrolle via App
  - App Starten + Anmeldung am Backend
  - Auswahl des Modus (Single, Multi, Spector)
  - Auswahl des Szenarios
  - Auswahl der Roboter
  - Start

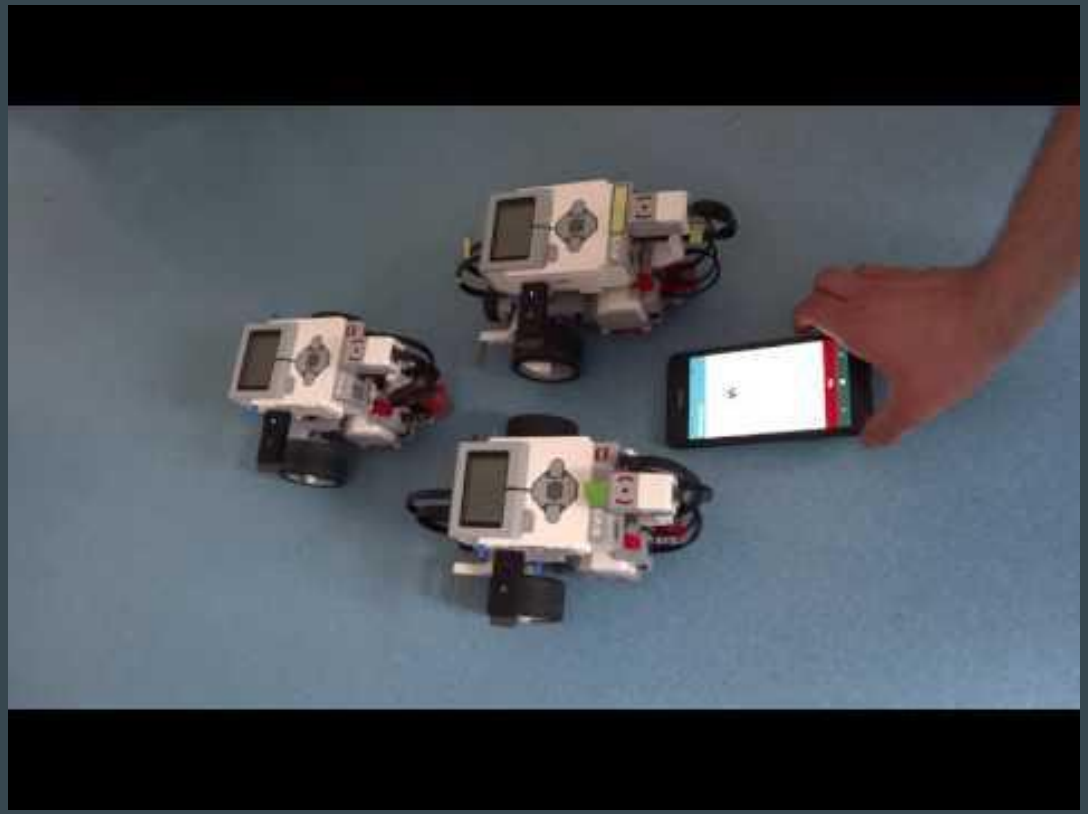
Herausforderungen

- Bluetooth
  - Nur Bluetooth Low Energy Unterstützung - Xamarin
  - Komplexe Implementierung - Java
- Verzögerte Kommunikation
  - Zeitversetzte Ankunft der Daten
  - Schwankende Latenz / Datenrate
    - Lösung:
      - TCP-No-Delay
      - Initialisierungs-Kommando
- LEGO - EV3 Roboter
  - leJOS Firmware
  - Unkontrolliertes / Unrepobuzierbares Verhalten (EV3 Java Library)

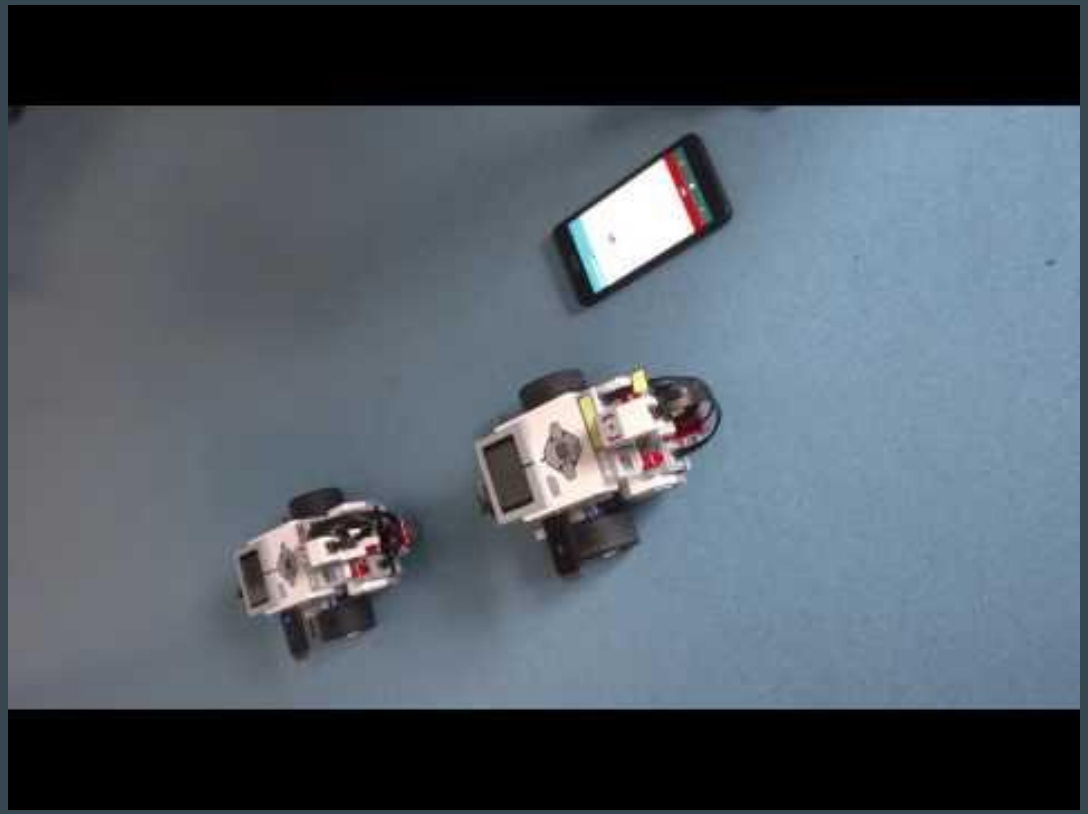
Ergebnis - Video



Ergebnis - Video



Ergebnis - Video



Ausblick

- Refactoring
- Erweiterung des Systems
  - Integration weiterer Sensoren
  - Implementierung weiterer Robotertypen
  - Umsetzung weiterer Szenarien
  - Multiuser-System

- Github - FleeAndCatch