DHBW Lörrach
Duale Hochschule Baden-Württemberg

NAO-Dashboard

Dokumentation

Projekt-Beschreibung Programmieren/Informatik-Labor

WiSe 17/18

Angewandte Informatik, 1. Semester

Dozent: Prof. Dr. Erik Behrends

Gruppenmitglieder:

Mustafa Mado, Khaled Jebrini, Michael Bachmann und Simon Bienroth

Umgesetzte funktionale MUST-HAVES:

- **Application/Session:**
  Verbindung mit NAO durch Angabe der IP-Adresse und des Ports. Die Daten zu Verbindungen werden auch nach Neustart in der GUI gemerkt und in einer Auswahlliste angezeigt. Es ist möglich, die Verbindung zu einem NAO zu trennen und sich zu einem anderen NAO zu verbinden.
- **ALMotion**:
  NAO ausruhen bzw. aufstehen lassen
- **ALRobotPosture**:
  Auswahl & Ausführung der verschiedenen NAO-Posen
- **ALMotion**:
  Fortbewegung und Drehung in verschiedene Richtungen (mit Buttons) inklusive Einstellung der Geschwindigkeit (durch einen Schieberegler/Slider).
- **ALMotion**:
  Ausrichtung des Kopfes (mit vier Buttons)
- **ALTextToSpeech**:
  Eingabe von Text, den NAO sagen soll (Deutsch/Englisch, Tonhöhe [„Pitch“])
- **ALLeds**:
  Festlegen der Farbe der Augen-LEDs (links und/oder rechts mit Farbauswahl)
- **ALBattery/ALBodyTemperature**:
  Akkuladestand/Temperatur des NAO anzeigen
- **ALAudioPlayer**:
  Abspielen vorinstallierter Sounds (der blaue NAO mit der IP ...127 liefert verschiedene Sounds mit audioPlayer.getSoundSetFileNames("Aldebaran"))
- **ALTouch**:
  Drei Berührungssensoren am Kopf mit konfigurierbarer Sprachausgabe belegen

Zusatz: Umgesetzte Großgruppen-MUST-HAVES:

- **ALMotion**: Bewegung der Arme und Hände
  - Steuerung über
    - Auswahl der Arme: links / rechts / beide
    - Auswahl der Gelenke: Schulter / Ellbogen / Hand (jeweils 2 Gelenke)
    - 4 Buttons für die Steuerung von jeweils 2 Gelenken in jeweils 2 Richtungen

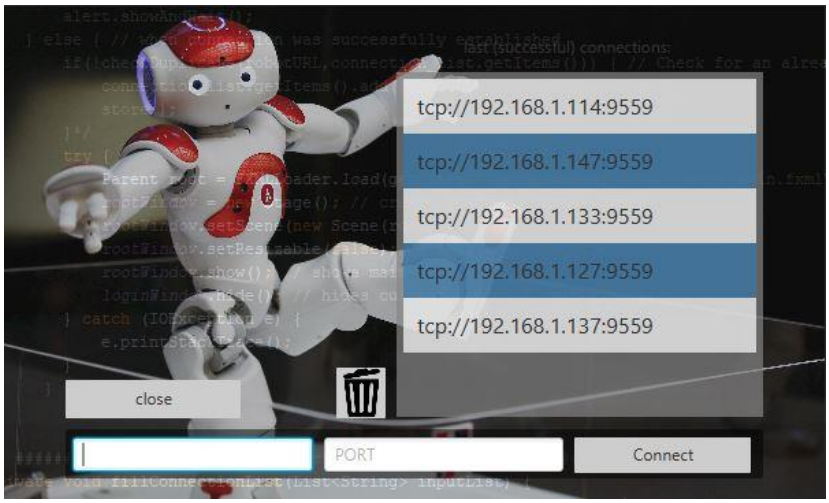
Umgesetzte nichtfunktionale Anforderungen:

- **GUI mit JavaFX** und eigener Style in CSS
- **Code lesbar** gestaltet, Kommentare an allen nicht-trivialen Stellen
- Soweit wie möglich: **einheitliche Code-Struktur**
- Ständige Arbeit mit **Github**
    - Wobei es zu sehr unterschiedlichen Commit Statistiken kam, da meistens zu zweit an einem Laptop programmiert wurde. So kann der zweite Laptop zum Recherchieren verwendet werden, ohne das Netzwerk zu wechseln.

Umgesetze NICE-TO-HAVES:

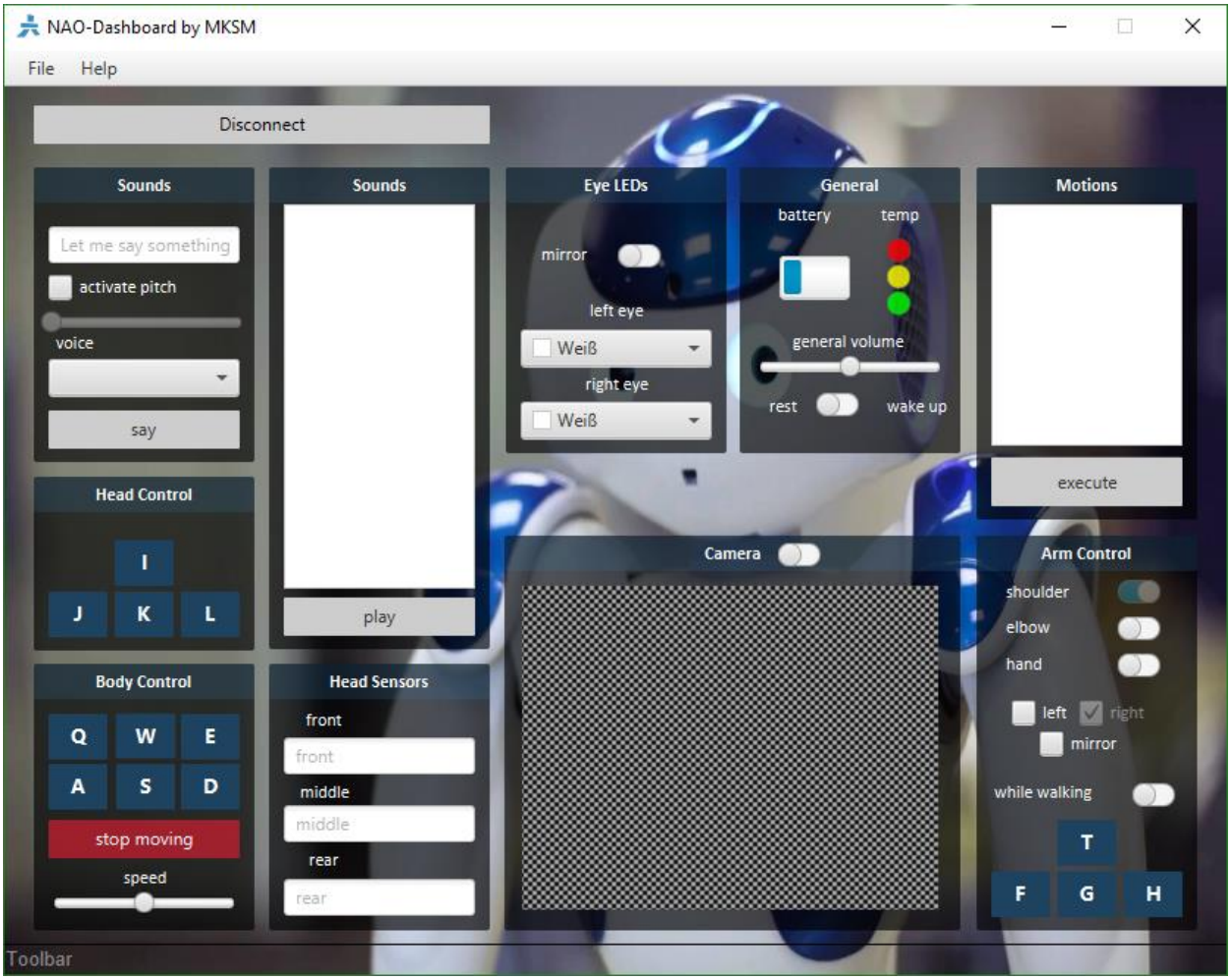
- **Application/Session**:
    - Verbindungsaufbau timeout im Falle einer falschen IP
    - Ständige Kontrolle des Verbindungsstatus und entsprechendes Fehlerhandling
- **ALMotion**: Fortbewegung und Drehung in verschiedene Richtungen (z.B. mit Knöpfen)
    - Bewegung der Arme während des Gehens kann ein- oder ausgeschaltet werden
- **ALTextToSpeech**: Eingabe von Text, den NAO sagen soll (Deutsch/Englisch, Tonhöhe)
    - Sprachauswahl findet über Stimmenauswahl statt.
- **ALVideoDevice**:
    - Live-View der Kamera in eigenem Thread um Performance Probleme zu vermeiden
    - Ein- und Ausschalten der Liveview (z.B. bei Performance Problemen durch schlechtem Netzwerk)

- **GUI**:
    - „Login“-(Connection) Window:



- speichert letzte (erfolgreiche) Verbindungen (keine Duplikate)
- Vermeidet falsche User-Eingaben
- Einzelne gespeicherte Einträge löschbar
- Verbindungen werden in .dashboard-File gespeichert
- Speicher/Lade-Funktion noch erweiterbar für andere Konfigurationen

- „Main-Window“:
  - Konsistentes, dynamisches Design (Modul, zB. Sounds erscheint nur, wenn auch verfügbar)
  - GUI-Elemente werden entsprechend gesperrt, wenn sie nicht vom User benutzt werden sollen



Abschlussbemerkung:

- Das Programm wurde hauptsächlich auf der vom Dozenten bereitgestellten Linux-VM getestet.
- In Moodle wurde keine .jar-Datei abgegeben, da die Erstellung einer (ausführbaren & lauffähigen) Datei trotz Video-Anleitung fehlschlug. Dies wurde am 21.02. ebenfalls mit dem Dozenten so abgesprochen.