**Ziel**

Das Ziel des Projekts war die Erstellung eines Frameworks zur Nutzung von Sensordaten von Androidgeräten auf javafähigen Endgeräten.

**Requirements**

Von Anfang an wollten wir mit unserer App 3 mögliche Anwendungen umsetzen: eine Maussteuerung, eine Spielsteuerung und eine Robotersteuerung. Für die Anwendungen war ein Requirement, dass die Verbindung sehr schnell und zuverlässig sein sollte. Gemessen haben wir dies an der Round-Trip-Time.  
Für die Maussteuerung hatten wir uns folgende Requirements gesetzt: Die Steuerung sollte sehr direkt sein, damit sie die Hauptaufgaben einer Maus übernehmen kann. Das sind Rechts- und Linksklick, damit verbunden der Doppelklick und das Markieren und bewegen von Dateien, sowie das Bedienen des Kontextmenüs.  
Für die Spielsteuerung haben wir einen SNES-Controller imitiert, mit dem wir das Spiel „Super Mario Kart“ spielen können. Unser Ziel war es auf der 1. Strecke mit der Gesamtzeit eines nativen Controller konkurrieren zu können. Nach einigen Tests ergab sich daraus eine Zeit von 1:20. Für den Mehrspielermodus brauchten wir dann eine Multiclientunterstützung. Um mehr Sensoren zu verwenden haben wir beschlossen die Item Mechanik durch einen weiteren Sensor zu steuern.   
Die Robotersteuerung ist eine Umsetzung des Kuka Roboters „KUKA LBR Iiwa 7 R800“, der ein Murmellabyrinth bedienen soll. Dazu wird er in eine bestimme Position gebracht, und es lässt sich nur das letzte Neigungs-Rotationspaar des Roboterarms bewegen. Dadurch ist es möglich das Labyrinth durch geschickte Neigungen zu lösen.  
Unser oberstes Ziel war es, die App und den Server sehr benutzerfreundlich zu gestalten. Die App sollte sehr leicht bedienbar sein, sodass sehr wenige Konfigurationen notwendig sind. Den App-Code selber muss man dafür nicht verändern, neue Funktionen sind am Server schnell hinzufügbar. So kann man z.B. das Layout für die Buttons, die vom Server gesetzt werden, durch eine XML-Datei anpassen. Für eine angenehme und genaue Steuerung sollte man durch einfache Datenfilter im Server die Sensibilität selber einstellen können. Die Server sollten natürlich auch leicht zu finden, und auch zu wechseln sein. Zur Benutzerfreundlichkeit gehört auch, dass die App Cross-Plattform einsetzbar ist. Unser Server ist auf allen Geräten mit JDK 8 und Zugang zu WLAN-Netzwerken zu nutzen, die App auf allen Android Geräten (Android 4.4 oder höher). Zum Verbinden muss man lediglich den Discovery Port einstellen, für die restlichen Verbindungen werden freie Ports automatisch gesucht.

**Lessons learned**

Es ist deutlich effektiver sich gemeinsam zu festen Zeiten zu treffen um zusammen zu arbeiten. Dadurch bekommt man direktes Feedback zu neuen Ideen und deren Umsetzungen von den anderen Teammitgliedern. Pairprogramming erlaubt Diskussionen und konstruktive Lösungen von neuen Problemen durch andere zu erhalten.  
Uns fiel es schwer messbare Requirements zu erstellen, die das Projekt beschreiben.