

Software Prototyping & Visualisierung

Übung 7+8 (Projekt)



Sommersemester 2014

Name:	<u>Richard Knoll, Florian Nairz</u>	Abgabetermin:	28.05.2014 14:40
Mat.Nr.:	<u>S1310567017, S1310567019</u>	Punkte:	<u> </u>
Aufwand in h:	<u>28</u>	korrigiert:	<u> </u>

Dieser Übungszettel ist in Zweiergruppen zu lösen. Es reicht, wenn ein Gruppenmitglied die Arbeit im Moodle abgibt.

Teil 1 (32 Punkte) Projekt - Applikation

Entwickeln Sie eine WPF Anwendung, die Sensordaten empfängt und verarbeitet. Sie haben freie Hand welche Sensoren Sie anbinden wollen. Sie können eine beliebige Schnittstelle verwenden, um auf die Daten zuzugreifen.

Entscheiden Sie sich für eine von 2 Möglichkeiten, was Sie mit den Sensordaten machen:

a) Visualisierung der Sensordaten

Stellen Sie die Sensordaten graphisch dar. Achten Sie je nach Sensor auf eine entsprechend sinnvolle Darstellungsform. Die Sensordaten sollen so zeitnah wie möglich dargestellt werden.

b) Interpretation der Sensordaten

Die Sensordaten sollen in der Form interpretiert werden, dass eine Aktion ausgelöst wird. Liefern sie zusätzlich auch ein Testprogramm, in dem man erkennt, was Sie steuern können/könnten.

Sie dürfen für diese Übung auch externe Libraries (z.B. aus Nuget) und fremden Code verwenden. Ziel ist es, einen funktionierenden Prototypen zu entwickeln.

Teil 2 (16 Punkte) Projekt - Präsentation

Zusätzlich zur Abgabe im Moodle müssen sie ihr Projekt auch in der Übung, am 28.05.2014 präsentieren. Jede Gruppe hat dafür 10min Zeit. Nutzen Sie diese 10 min nicht nur um die Funktionalität ihrer Applikation zu präsentieren, sondern auch, um die interessantesten Code-Passagen zu erklären.

Sie können für Ihre Präsentation maximal 16 Punkte erhalten.

Allgemeine Hinweise: Legen Sie bei der Erstellung Ihrer Übung großen Wert auf eine **saubere Strukturierung** und auf eine **sorgfältige Ausarbeitung**! **Dokumentieren** Sie alle Schnittstellen und versehen Sie Ihre Algorithmen an entscheidenden Stellen ausführlich mit **Kommentaren**! **Testen** Sie ihre Implementierungen ausführlich! Geben Sie **Lösungsideen** an!

1 Teil 1

1.1 PresIt

In dieser Übung wurde ein Präsentationssystem entwickelt und als Prototyp implementiert.

Ziel dieses Systems ist es, eine einfache Verwaltung von Präsentationen zu ermöglichen und diese auf dem Präsentationsrechner zu präsentieren. Dabei soll sie mit externen Geräten bzw. Sensoren gesteuert werden können. In diesem Fall ist das ein Beschleunigungssensor, womit eine Gestensteuerung ermöglicht wird.

Das System besteht aus insgesamt 4 Komponenten, die miteinander kommunizieren und so die Steuerung und Verwaltung von Präsentationen ermöglicht. Im Folgenden werden die einzelnen Komponenten näher erklärt.

1.1.1 Server

Hier sind die Präsentationen gespeichert. Realisiert wurde er mit einem WCF Service, der einfach in einen IIS integriert oder Standalone betrieben werden kann. Alle Clients verbinden sich zu diesem Server, der anhand einer privaten ID des Smartphones eine Zuordnung der Präsentationen trifft. Dadurch ist ein Server in der Lage, mehrere Präsentationen und Benutzer gleichzeitig zu verwalten. Die Präsentations Clients können, nach einem Login mit dem Smartphone, alle erstellten Präsentationen ändern, neue anlegen oder eine starten. Dann wird auf Befehle des Android Clients gewartet, der die Kommandos an die Präsentation senden kann, wie zum Beispiel „Nächste Folie“.

1.1.2 Präsentations Client

Der Desktop Rechner, der die Präsentation typischerweise auf einem Projektor anzeigen soll, verbindet sich über die Windows Applikation zum Server und muss zunächst von einem Smartphone freigeschaltet werden. Diese Freischaltung erfolgt über das Scannen des angezeigten, zufällig generierten QR Codes, der die Session ID darstellt. Nach einem Login können die Präsentationen verwaltet und gestartet werden. Zusätzlich zur Steuerung mittels Smartphone, kann auch per Tastatur weiter und zurück geschaltet werden.

1.1.3 Android Client

Auch auf dem Android Client konnte als Sprache C# verwendet werden. Dafür wurde das Xamarin Framework verwendet, und erlaubte es, eine gemeinsame Bibliothek für alle Clients und den Server zu verwenden. Auf dem Smartphone ist eine private ID gespeichert, die nach dem ersten Start der App generiert wird. Dieser dient als Zugangsschlüssel zu allen Präsentationen des Besitzers. Die Präsentationen können entweder mit den Tasten auf dem Bildschirm, oder per Gesten gesteuert werden, die allerdings zunächst trainiert werden müssen. Dafür können entweder die Sensoren des Smartphones, oder ein per Bluetooth angeschlossenes Device verwendet werden. Nachdem 3 Gesten eingelesen wurden (Zurück, Vorwärts, Pause), können diese sofort verwendet werden, um die Präsentation zu steuern.

1.1.4 Embedded Device

Um ein externes Gerät zu veranschaulichen, wurde ein embedded Device verwendet, das bereits Beschleunigungssensoren auf der Platine besitzt. Zusätzlich wurde ein Bluetooth Modul angeschlossen, um die Daten drahtlos an das Smartphone übertragen zu können. Dieses Gerät gibt

die Daten des Sensors unverändert an die C# App auf dem Smartphone weiter, das auf diesen Daten die Gestenerkennung ausführt.

1.1.5 Gestenerkennung

Die Gestenerkennung erfolgt auf dem Smartphone direkt auf den Rohdaten. Eingesetzt wurde der "Dynamic Time Warping"-Algorithmus, der nach passenden Patterns in einem dynamischen Datenstream sucht und diese bewertet. Sobald eine hohe Übereinstimmungsrate zu den zuvor trainierten Bewegungen gefunden wurde, wird die entsprechende Aktion ausgelöst.

1.2 Screenshots

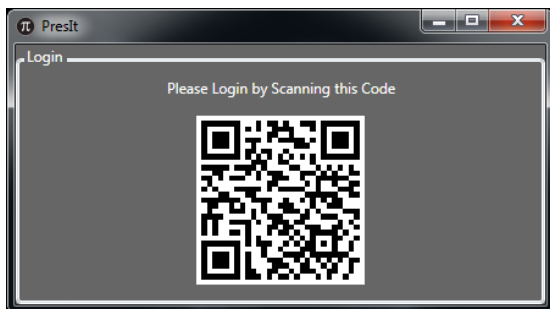


Abbildung 1: GUI nach dem Start, warte auf Login per Smartphone

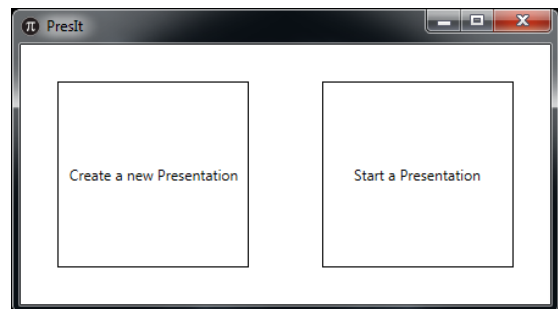


Abbildung 2: GUI nach erfolgreichem Login

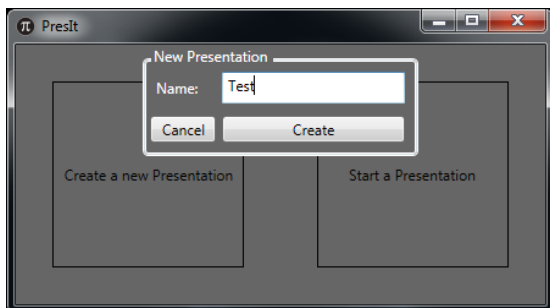


Abbildung 3: Anlegen einer neuen Präsentation

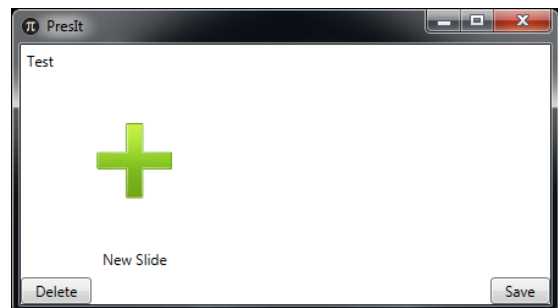


Abbildung 4: Neue Folien per Drag and Drop ins Fenster ziehen



Abbildung 5: Fertige Präsentation

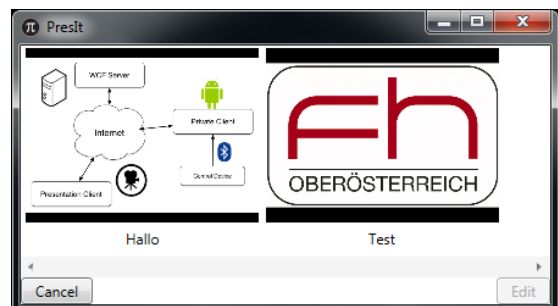


Abbildung 6: Auswahl einer Präsentation,
die gestartet werden soll

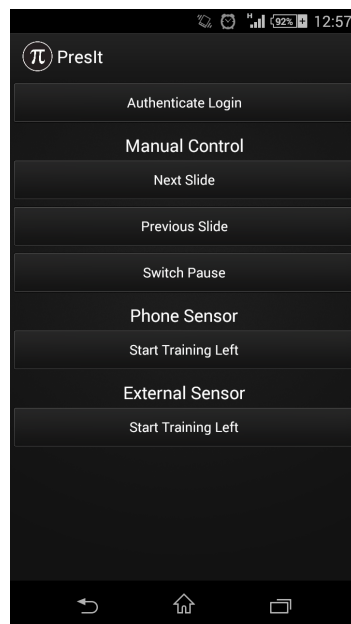


Abbildung 7: Smartphone App

2 Teil 2

2.1 Präsentation

Die Präsentation und alle Sourcen sind in der Abgabe, zusätzlich wurde das Projekt auf GitHub online verfügbar gemacht.

Außerdem ist eine kleine Demonstration auf YouTube zum nachträglichen Betrachten bereitgestellt.

Sources: github.com/rknoll/presit

Video: youtu.be/OgyndkuS0jU