

SOFTWARETECHNIK PRAKTIKUM

Recherchebericht
Sommersemester 2017

Projekt: ENA-17: Big Data & KINECT - Eingangsassistent

Vladimir Moroz, Christian Fuß, Marco Neumann,
Patrick Oswald, Sven Oswald, Chiara Hergl

Inhaltsverzeichnis

1	Begriffe	3
1.1	Xbox 360	3
1.2	Kinect	3
1.3	Xbox 360 Development Kit	3
1.4	Big Data	3
1.5	Webserver	4
1.6	Server-Client Struktur	4
1.7	Front-End	4
1.8	Back-End	4
1.9	Softwarelizenz	4
2	Konzepte	5
2.1	J4K	5
2.2	JNI	5
2.3	Java Maven	5
2.4	TDD	6
2.5	Java JUnit	6
2.6	Bootstrap - adminLTE	6
3	Aspekte	7
3.1	Einleitung	7
3.2	Ziel	7
3.3	Vergleichbare Projekte	8
3.4	Probleme	8
4	Literaturverzeichnis	9

1 Begriffe

1.1 Xbox 360

Die Xbox 360 [1] ist eine stationäre Spielekonsole der 7. Konsolen-Generation, welche im Jahr 2005 von Microsoft erstmals verkauft wurde. Sie ist der Nachfolger der Xbox (2002) und der Vorgänger der Xbox one (2013).

1.2 Kinect

Bei einer Kinect [2] handelt es sich um eine Hardware zur Steuerung der Xbox 360. Diese wird über Körperbewegungen, sowie Sprache gesteuert. Die Eingabe wird dabei von der Kinect über Prime Sense-Tiefensensoren, 3D-Mikrofone und Farbkameras erkannt und von der entsprechenden Software verarbeitet.

1.3 Xbox 360 Development Kit

Ein Xbox 360 Development Kit [3] (kurz: DevKit) ist ein Entwicklungswerkzeug der Spieleentwicklung. Es wird außerdem zum Testen, sowie Debuggen benutzt.

1.4 Big Data

Big Data [4] wird als Sammelbegriff für digitale Technologien verwendet, bei denen große digitale Datenmengen, sowie deren Analyse, Sammlung, Verwendung und Vermarktung im Mittelpunkt stehen. Dabei werden 5 Big Data Challenges gestellt, welche sich in 2 Teilbereiche aufteilen. Es handelt sich um volume (bis zu viele Peta- und Exabyte an Speicher), velocity (sofortige Analyse dynamischer Datenströme) sowie variety (hohe Qualität und Glaubwürdigkeit der Daten), welche unter den Begriff 'Big' fallen, sowie value (Gewinnung nützlicher Informationen) und variety (he-

terogene strukturierte, teil- und unstrukturierte Daten), welche für die Sicherstellung der Daten verantwortlich sind.

1.5 Webserver

Ein Server ist ein zentraler Computer, mit dem sich andere Computer verbinden können, um bereitgestellte Dienste/Daten zu erhalten. Bei Webservern [5] handelt es sich um Server, die entweder für die Öffentlichkeit bestimmt sind und als WWW-Dienst eingesetzt werden oder sie werden lokal im internen Netz eingesetzt.

1.6 Server-Client Struktur

Die Server-Client Struktur [6] ist eine Möglichkeit der Aufgabenverteilung im Netzwerk. Dabei stellt der Server Daten/Dienste bereit und wartet auf Anfragen, um diese zu verschicken/leisten. Der Client hingegen verbindet sich mit dem Server, um Dienste anzufragen.

1.7 Front-End

Front-End [7] ist ein Begriff der Schichteneinteilung aus dem Bereich der Informationstechnik. Es handelt sich dabei um den Teil der Software (hat jede Software), die von dem Nutzer gesehen wird und mit der dieser agieren kann, wie zum Beispiel die Nutzeroberfläche einer Webseite.

1.8 Back-End

Back-End [7] ist ein Begriff der Schichteneinteilung aus dem Bereich der Informationstechnik. Es handelt sich dabei um den Teil der Software, in dem die Funktionen dieser festgelegt werden. Das Front-End ist abhängig von diesem Bereich, da die Definition dieses in jenem Bereich erfolgt. Dabei ist das Back-End für den Nutzer nicht sichtbar.

1.9 Softwarelizenz

Programme, welche im öffentlichen Rahmen ausgeführt werden, benötigen eine Nutzungserlaubnis. Diese Nutzungserlaubnisse werden urheberrechtliche Softwarelizenzen [8] genannt.

2 Konzepte

2.1 J4K

Java for Kinect [9] ist eine OpenSource Java Bibliothek, welche die Programmierung der Microsoft Kinect zu ermöglichen. Sie ist kompatibel mit allen Kinectgeräten. Dafür wird eine Verbindung zu den Sensoren der Kinect aufgebaut und die Signale ausgewertet.

2.2 JNI

Java ist keine plattformunabhängige Programmiersprache. Um plattform-spezifische Funktionen aufzurufen, wird die standardisierte Anwendungsprogrammierschnittstelle **Java Native Interface** [10] benutzt.

2.3 Java Maven

Maven [11] ist ein Tool, um die Wiederverwendung von Java-Bausteinen zu fördern und die Entwicklung zu vereinfachen. Es steht dabei in direkter Konkurrenz zu Ant, hat jedoch weitaus mehr Funktionen, als das einfache Build-Tool. Es dient zum einen dazu einen Build-Prozess zu erstellen, der die Erstellung der Executables automatisiert und vor allem bei größeren Projekten vereinfacht. Maven bietet dabei die Möglichkeit einer einfachen Einbindung von eigenen oder fremden Bibliotheken. Build-Tools im allgemeinen, jedoch Maven im speziellen können folgende Aufgaben erfüllen:

- Downloaden von Abhängigkeiten (dependencies)
- Compilen des Sourcecodes zu Binaries
- Packaging der Binaries
- Tests ausführen

- Tests automatisieren
- Deployen der Binaries als produktives/lauffähiges Gesamtsystem

2.4 TDD

Test Driven Development [12] ist eine Methode der Programmierung, welche bei der Entwicklung von Software eingesetzt wird. Dabei werden die Tests für die Software vor der eigentlichen Software geschrieben. Dies hat den Vorteil, dass es nicht, wie bei anderen Methoden, zu mangelnder Testbarkeit kommt oder das Programm aus Zeitgründen nicht ausführlich getestet werden kann, sowie kommt es nicht so leicht zu Redundanzen.

2.5 Java JUnit

Um die Funktionalität einer Software oder einem Teil davon sicherzustellen, bieten sich unit tests an. Dabei handelt es sich um kleine Programme, welche ohne Benutzerkontrolle den Code automatisch auf bestimmte, vorher definierte Regeln zu testen. JUnit [13] stellt dabei ein solches Testframework dar.

2.6 Bootstrap - adminLTE

AdminLTE [14] basiert auf Bootstrap, wobei Bootstrap ein Open Source Framework ist, welches auf allen mobilen Geräten läuft und zugleich sehr anpassungsfähig ist. Dabei liefert Bootstrap eine umfangreiche Dokumentation für HTML, sowie CSS, sowie JavaScript Erweiterungen.

3 Aspekte

3.1 Einleitung

Im Big Data Kompetenzzentrum an der Universität Leipzig (ScaDS (Dresden/Leipzig)) werden aktuelle Big Data Technologien zur Sammlung, Speicherung und Analyse von sehr heterogenen und großen Datenmengen evaluiert. Insbesondere Video- und Bewegungssensordaten von Personen sind in vielen Bereichen von sehr großem Wert. Zur Evaluation zum Kennenlernen vorhandener Technologien im Umgang mit solchen Daten soll im Praktikum ein einfacher Big Data Eingangsassistent entwickelt werden, der kommuniziert, Daten sammelt und auf Handlungen und Gesten von Personen reagieren kann. Dazu wollen wir mittels KINECT Daten sammeln und in einem verteilten Key-Value Store (Apache Accumulo) zeitbezogen speichern und analysieren.

3.2 Ziel

Im SWT-Praktikum soll durch Studenten ein einfacher Eingangsassistent entwickelt werden. Dieser besteht aus

1. einem Client zur Kommunikation auf Basis einer Text-2-Speech Engine
2. einem web-basierten Datenverwaltungs-, Speicherwerkzeug auf Basis von Apache Accumulo sowie
3. einer Softwarekomponente zur Anbindung einer KINECT zur Aufnahme von Nutzerverhalten und Gesten.

Als Programmiersprache sollte Java + HTML5 verwendet werden. Innerhalb eines agilen Softwareentwicklungsprozesses (SCRUM) arbeiten die Studenten kooperativ, teilen sich Aufgaben auf und entwickeln gemeinsam einen funktionalen Prototypen.

3.3 Vergleichbare Projekte

Es existieren viele Projekte, welche sich mit dem Identifizieren von Personen mit Hilfe der Kinect beschäftigen. Eines wird in „Real Time Person Tracking and Identification using the Kinect sensor“ [15] beschrieben, welches an dem Worchester Polytechnic Institute entstanden ist. Dabei werden Sprachbefehle zur Steuerung von Smart Homes benutzt.

Der Unterschied zu unserem Projekt liegt in der Verarbeitung der Sprache. In dem hier beschriebenen Projekt, handelt es sich um ein Mittel zur Steuerung. Bei uns hingegen soll sie zu Text übersetzt werden und an einen Server beziehungsweise eine andere Person übermittelt werden, welche darauf eingehen kann.

Außerdem spielt in unserem Projekt das Sammeln von Daten eine entscheidene Rolle, was in dem hier Beschriebenen nicht behandelt wird.

3.4 Probleme

Die Sensoren der Kinect v2, mit welcher wir arbeiten, sollen ziemlich genau sein. Trotzdem wird es ein Problem sein, die Daten, welche wir erhalten auszuwerten und genau zuzuordnen. Dies fängt bei groben Unterscheidungen, wie von Menschen und sich bewegenden Gegenständen, an und reicht hin bis zu Feineren, wie die Unterscheidung von verschiedenen Personen, wobei Lichtverhältnisse auch beachtet werden müssen.

Außerdem muss beachtet werden, dass es sich um mehrere Personen vor dem Sensor handeln kann und wie damit umgegangen wird.

Ein weiteres Problem wird der Umgang mit einzulesender Sprache, da nicht alle Menschen gleich sprechen und somit eine gewisse Toleranz in der Verarbeitung vorhanden sein muss, die jedoch nicht zu groß sein darf, da ansonsten viele Fehler entstehen könnten. Dafür muss eine ausreichend genaue 'Speech to Text' API gefunden werden, mit der wir arbeiten können.

Keiner aus unserem Team hat bisher Erfahrungen mit dem Programmieren der Kinect gesammelt und da wir diese nur im BigData Kompetenzzentrum zur Verfügung gestellt bekommen, müssen wir eine Möglichkeit finden unseren Code auch ohne diese zu testen.

4 Literaturverzeichnis

- [1] https://de.wikipedia.org/wiki/Xbox_360
- [2] <https://de.wikipedia.org/wiki/Kinect>
- [3] https://de.wikipedia.org/wiki/Xbox_360#Development_Kits
- [4] https://de.wikipedia.org/wiki/Big_Data
- [5] <https://de.wikipedia.org/wiki/Webserver>
- [6] <https://de.wikipedia.org/wiki/Client-Server-Modell>
- [7] http://praxistipps.chip.de/backend-und-frontend-was-ist-das-einfach-erklaert_41384
- [8] <https://de.wikipedia.org/wiki/Lizenz#Softwarelizenzen>
- [9] <http://research.dwi.ufl.edu/ufdw/j4k/>
- [10] <http://docs.oracle.com/javase/6/docs/technotes/guides/jni/>
- [11] https://de.wikipedia.org/wiki/Apache_Maven
- [12] https://de.wikipedia.org/wiki/Testgetriebene_Entwicklung
- [13] <https://de.wikipedia.org/wiki/JUnit>
- [14] <https://almsaeedstudio.com/>
- [15] <https://web.wpi.edu/Pubs/E-project/Available/E-project-042513-081838/>