BACHELOR PAPER

Term paper submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Engineering at the University of Applied Sciences Technikum Wien - Degree Program TW

Integrating Assistive Technology (Working title)

By: Leonhard Hauptfeld

Student Number: 1510768031

Supervisor: Ing. Martin Deinhofer, MSc

Vienna, January 16, 2018

FH University of Applied Sciences
TECHNIKUM WIEN

Declaration

“As author and creator of this work to hand, I confirm with my signature knowledge of the relevant copyright regulations governed by higher education acts (see Urheberrechtsgesetz /Austrian copyright law as amended as well as the Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I hereby declare that I completed the present work independently and that any ideas, whether written by others or by myself, have been fully sourced and referenced. I am aware of any consequences I may face on the part of the degree program director if there should be evidence of missing autonomy and independence or evidence of any intent to fraudulently achieve a pass mark for this work (see Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I further declare that up to this date I have not published the work to hand nor have I presented it to another examination board in the same or similar form. I affirm that the version submitted matches the version in the upload tool.“

Vienna, January 16, 2018 Signature

Kurzfassung

In einer sich rapide entwickelnden Welt entstehen täglich neue experimentelle Softwaretechnologien. Manche davon etablieren sich, manche sind schnell veraltet. Durch den Nutzen dieser Neuerungen in dem Feld der Assistiven Technologien bieten sich jedoch große Möglichkeiten an. Dafür müssen neue Komponenten aber aufgrund der schnellen technischen Entwicklung ebenso schnell in existierende Assistive Toolkits integriert werden. Diese Arbeit beschäftigt sich aus diesem Grund mit dem Prozess der Integrierung neuer Software in ein AT-Toolkit. Dabei werden zuerst verschiedene Methoden zur Bewertung und Integrierung von Software in einem Assistive Technologies-Umfeld besprochen und bewertet. Dies inkludiert eine Methodik zur Analyse von Quell- und Zielsoftware inklusive Erkennung von Gemeinsamkeiten sowie Erweiterbarkeitskonzepten. Aus den Ergebnissen dieser Analyse wird ein Konzept der Implementierung erstellt. Weiters werden mehrere State-of-the-art Technologien wie die Programmiersprachen Java, C++, Python und die Datenaustauschformate XML und JSON anhand ihrer Interoperabilität und Nutzen im assistiven Umfeld untereinander gegenübergestellt und bewertet. Anschließend werden die Ergebnisse im Zuge einer Integrierung von Gestenerkennung mit Intel RealSense-Technologie in das AT-Toolkit "Assistive Technology Rapid Integration & Construction Set" angewendet. Abschließend wird die Effektivität der Methodik anhand der erfolgten Implementierung bewertet und auf positive bzw. negative Aspekte hin analysiert. Aufgeteilt nach Anwendungsfällen werden die Methoden diskutiert und bewertet sowie "Best Practices" aufgelistet.

Schlagworte: Assistive, Technologie, Software, Integration, Einbindung, Bewegungserkennung, Computer Vision

Abstract

In today's rapidly advancing world, new software technologies emerge at an enormous pace. Some may develop to be the next great standard, while others perish into obscurity rapidly. The uses for these emerging technologies in the field of assistive technology are often manifold but the rapid pace of technology demands fast adaption of this software to existing toolkits, or they themselves might soon disappear. This paper brings up the key challenges involved with this implementation and adaption process and possible methods for solving them. First, different processes pertaining to the analysis and integration of software in the context of assistive technologies are analyzed, discussed and rated. This includes workflows with the purpose of analysing source and target software, collecting data such as similarities and expansion concepts. Furthermore, state-of-the-art technologies such as the programming languages Java, C++ and Python as well as the data exchange formats XML and JSON are weighed against each other and rated for their usage in assistive technologies. The results of these discussions and the methods discussed are then applied to an integration of gesture recognition utilizing Intel RealSense technology into the AT-Toolkit "Assistive Technology Rapid Integration & Construction Set", or "AsTeRICS". The paper concludes with a rating of effectiveness as well as positive and negative aspects for the discussed methods, using the aforementioned implementation as a guideline. Sorted by application, the different methods and processes are discussed and a lists of "best practices" are developed.

Keywords: assistive, technology, software, integration, motion recognition, computer vision

Acknowledgements

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Contents

1 Introduction

1.1 State of the art

1.1.1 AT-Frameworks

AsTeRICS is a framework for building AT-Solutions that is based entirely on different plug-ins and their interaction. Possible plug-in types are sensors and actors. It is written in Java, utilizing native C++ libraries where necessary or advantageous.

1.1.2 Computer Vision

1.1.3 Programming languages

Java

Java[2] is a programming language invented by Sun Microsystems, now owned by the Oracle Corporation, which also supports its most prominent implementation. It compiles into non-native Java Bytecode that runs on a Java Virtual Machine (JVM). As a result, any compiled Java Code can run on any hardware that runs such a JVM, making Java almost entirely platform independent.

C++

C++ is a programming language defined and standardized by the International Organization for Standardization (ISO)[1]. There are multiple implementations of this standard (each differing slightly), the most prominent ones being as part of the free GNU Compiler Collection project and Visual C++ by Microsoft.

Python

Python[3] is an interpreted programming language with a simple syntax, partially derived the language "ABC", a simple language with the original purpose of teaching children programming. Despite its simple appearance it is closely tied in with C and C++, with many libraries being direct ports from these programming languages. Prominent examples of this include OpenCV, QT and many more.

This versatility coupled with the easy syntax makes Python a very attractive technology to use for prototyping and scientific computation.

1.1.4 Data exchange formats

1.2 Combination of existing technologies

2 Method

2.1 Prerequisites

2.1.1 Utilized software

2.1.2 Required hardware

2.2 Implementation

2.2.1 Implementation of Handtracking

RealSense and OpenCV

Gesture recognition

2.2.2 Integration into framework

Creation of AsTeriCS plugin

Java Native Interface

3 Results

3.1 Technology integrated

3.2 Usage of Asterics

3.2.1 Plugin Configuration and Options

3.2.2 Tracking and Information Window

4 Discussion

4.1 Steps for approaching integration

4.1.1 Analyzing extensibility concepts

4.2 Combining different programming languages

4.2.1 Virtues and choosing languages

4.2.2 Difficulties and compensation

Bibliography

[1] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *C++ language specification*.

[2] ORACLE CORPORATION: *Java, Official Website*.

[3] PYTHON SOFTWARE FOUNDATION: *Python Programming Language, Official Website*.

List of Figures

List of Tables

List of Code

List of Abbreviations

ABC Alphabet

WWW world wide web

ROFL Rolling on floor laughing

A Anhang A

B Anhang B