

# Ing. Jelle Meeus

## Software Developer

Last update: September 2, 2022

The online version is available at

<https://jellemeeus.github.io/JelleMeeus.github> The

online version is available at

<https://jellemeeus.github.io/JelleMeeus.github>

### Location

2580 Putte

### Email

[jelle.meeus@hotmail.com](mailto:jelle.meeus@hotmail.com)

### Language

Dutch (native), English (fluent), French (intermediate)

### Driver's license

B

### Website

<https://jellemeeus.github.io>

Python	+++	Git	+++	Javascript	++	VIM	++	Java	++	CI/CD	++
C/C++	++	PL/SQL	++	8051	++	React	++	C#	++	Docker	+
				ASM							

## Education

Master in de industriële wetenschappen, elektronica-ICT 2018

*Katholieke Universiteit Leuven - Campus De Nayer - Sint-Katelijne-Waver*

## Thesis - Continuous Unobtrusive User Authentication Using Gait For Wearable Devices, Utilising Machine Learning Algorithms

[Text](#) [PDF](#) [Powerpoint Slides](#) [PDF](#)

In recent years, much research has been done to find new authentication methods that try to avoid explicit input from a user. These techniques use patterns and biometrics from a user to recognize machine learning models. One of these biometrics is the way a person walks. It can be captured by sensors on a smartwatch or smartphone, easily and unobtrusively. I researched a new feature-extraction technique to allow traditional machine learning models to recognize walking activity and recognize gait in a fast and accurate way. I developed a server application (Python) to process gait data and train models. I implemented a wearable application (Android) to record data and run inference of said trained models in real-time.

[Machine Learning](#) [Biometrics](#) [AI](#) [Python](#) [Java](#) [Android](#)

## 8051 Microcontroller Instruction Set IEEE754 32bit Floating-Point Library

Implementation of a IEEE754 Floating-Point library on the aduc832 system platform for 8bit 8052 based systems. The library allows for correct and fast calculation of the 4 basic operations (+, -, \*, /) for two numbers in IEEE754 32b without using MUL/DIV instructions.

[source](#) [ASM](#) [8051](#) [aduc832](#) [IEEE754](#)

## Work Experience

### Machine Learning Algorithms Student

[OneSpan](#) 4 weeks summer 2017

Developed a demo allowing recording and analysis of gait data on an Android wearable device

[Machine Learning](#) [Biometrics](#) [AI](#) [Python](#) [Java](#) [Android](#)

### Junior Software Engineer Consultant

[Sioux Embedded Systems](#) Nov 18 - Feb 19

I wrote tools accommodating a customer's migration from a Perforce Version Control System (VCS) to Git. I implemented existing and new features using Python and Gitlab API to meet developer demands. I setup a R&D

Internal website (Bootstrap) to provide an overview of projects that updates nightly. I did smaller IT related tasks, such as setting up automated backups of firewall settings and Jenkins configurations. Also, I wrote some automated tests in an in-house testing framework.

[Python](#) [Git](#) [Perforce](#) [Docker](#)

## Open Source Contributions

### [Azerothcore](#)

Complete Open Source and Modular solution for MMO

[source](#) [contributions](#) [C++](#) [SQL](#) [mmorpg](#) [emulator](#) [game](#)

## Hobby projects

### [Drawing Cards](#) [Demo Here](#)

A card drawing web app created with React hosted in github pages. Create and interact with a deck of French-suited SVG rendered cards. You can draw one or multiple cards, shuffle, flip over the deck. All neatly displayed through React with a status bar, menu bar and mouse over interaction.

[source](#) [React](#) [Javascript](#)

### [Bomberman](#) [Play Here](#)

Example of a 2D Classic Bomberman game made with godot. You can play as a bomb laying bunny and walk around a maze to reach a carrot and try not to blow yourself up in the process.

[source](#) [C#](#) [Godot](#)

### [Twitch Compilations From Cluster Data](#)

Create Twitch compilations and upload to Youtube with ease. Find clips by creators, clip ids, clip urls, game ids, category name, or a cluster based on Twitch Atlas.

[source](#) [Python](#) [Javascript](#) [React](#) [ElectronJS](#) [Twitch API](#) [Youtube API](#)

### [Automated Local And Cloud Backups With Cronjobs](#)

Easy automated backups to local and remote drives with cronjobs, rsync and rclone. We can specify which files to upload with filters (\*.txt) and easily upload to multiple cloud drives with variable data cap limits.

[source](#) [crontab](#) [GNU/Linux](#) [rsync](#) [rclone](#)

### [Home Media Server](#)

docker-compose.yml for a Home Media Server stack including: transmission (+ openvpn), jackett, radarr, sonarr, lidarr, calibre, calibre-web, plex, soulseekqt

[source](#) [Docker](#)