

# Ing. Jelle Meeus

## Software Developer

Last update: March 13, 2024

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.be>

### Github

[jellemeeus](https://github.com/jellemeeus)

Python	+++	Git	+++	Java	++	C/C++	++	C#	++	.NET	+
Lua	++	Typescript	++	Javascript	++	React	++	Vue	++	8051	++
SQL	++	MongoDB	++	CI/CD	++	VIM	++	VSCode	++	Docker	+

## 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](#)

I researched a gait-based authentication method amidst growing interest in non-explicit user input. I enhanced an existing barebone implementation, featuring an Android wearable app for data recording and a Python server app for offline processing. I integrated human activity and gait recognition systems, employing traditional machine learning models and a novel, fast, and accurate feature extraction technique. The outcome is a seamless, continuous gait-based authentication system, enabling offline data capture, server-based training, and real-time evaluation on a wearable.

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

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

Implementation of an IEEE754 Floating-Point library for aduc832 system platform for 8-bit 8052 based systems.

This library allows for accurate and fast calculations of the four basic operations (+, -, \*, /) for two numbers in IEEE754 32-bit floating-point format 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](#)

### 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

## Open Source Contributions

### [Azerothcore](#)

Contributed to Azerothcore project, a Complete Open Source and Modular solution for MMOs, by actively submitting pull requests addressing bugs in their Core (C++) and database (SQL). Includes debugging, troubleshooting by identifying and resolving issues in the codebase. Additionally, aided in quality assurance by testing and providing feedback on PRs from other contributors.

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

## Hobby projects

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

A card drawing web app created with React hosted on 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](#)

### [Words Vocabulary + Pronunciation + Definition](#)

Easy-to-read and printable vocabulary lists for commonly used English words. The lists cover spelling, meaning, example usage, and lexical spelling. The formatted lists are ordered alphabetically, by CEFR rating, and randomly. They are accessible in pdf and html formats.

[source](#) [Python](#) [Jupyter Notebook](#) [CI/CD](#) [Document Generation](#)

### [Server Status Monitor With Alerts](#)

Continuous Discord bot, written in Python, utilizing the Blizzard API to monitor server status and triggering ping alerts in a designated channel upon unlock status

[source](#) [Python](#) [Discord bot](#) [Blizzard API](#)

### [Loot Addon](#)

Loot distribution addon that allows for parallel auctioning

[source](#) [Lua](#) [WoW API](#)

### [Auction](#)

Web app were you can create a room, invite others and bid on auctions simultaneously in real time. Export and import ingame auctions with FLoot addon

[source](#) [Vue](#) [Typescript](#) [Javascript](#) [dotnet](#) [C#](#) [Lua](#) [MongoDB](#) [WoW API](#)