# Portfolio

*A few*

## Personal and Academic Works

Email: o.lemonakis@live.dk

Phone: +45 31444686

# Bachelor’s Thesis

## Exploring Vision-Based Interactions in VR Horror Games

When looking for an area of study for our bachelor’s thesis, we sought out the proposed fields of study and bachelor’s projects by faculty professors. We came upon Ken Pfeuffer and his proposals concerned ubiquitous computing and the research of AR and VR applications utilizing eye-tracking. We discussed some of the proposed projects with Ken, but he quickly sniffed out that we had both an interest and experience with game development. As such he proposed we could build a game for our study environment, and conduct research in such an environment. After much deliberation, we came upon the idea of attempting to quantify the user’s experience with vision-based interactions, and how effective different interactions would be in intensifying the experience for the user. We would build this environment in Unity, with the Meta Quest Pro headset.

Early on however, we realized that with the limited libraries and SDK’s available for eye-tracking development in VR, specifically for Unity, and with the scarcity of available eye-tracking capable headsets, we settled on imitating gaze interactions, and settling on a new term for this imitation, “vision-based interactions”. We built and tested the final environment with the Meta Quest 2 instead and forwent eye-tracking.

We built the environment as a horror game and sought to measure how scary these different types of interaction were for the users.

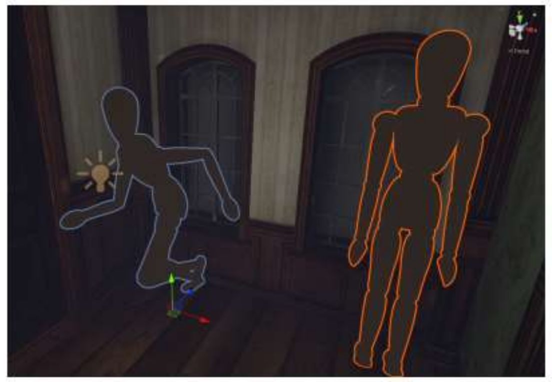
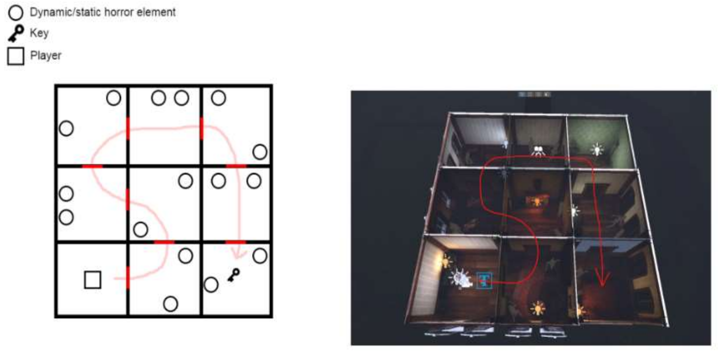


Figure 11: The dynamic horror element in the shape of a mannequin in different poses




(a) Diagram of the level layout

(b) Implementation in Unity of the level layout

Figure 8: Example of one of the 12 levels in the prototype

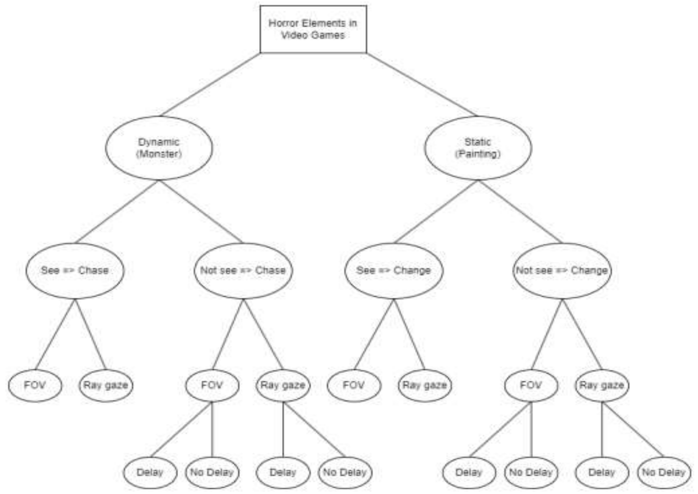



Figure 13: Interaction Tree

Figure 10: One of the rooms of the prototype using baked GI + AO and custom

Our results suggest that maybe eye-tracking isn’t necessary for gaze type interactions and that certain interactions are stronger than others.

Excerpt from the paper's abstract:

*"Gaze interaction in video games, using eye tracking, is in growth. No longer only used for novelty interaction but also as an integral part of the game’s mechanics. However, while gaze interaction is becoming more and more prevalent, gaze aversion appears to continue in novelty. Within this paper, we explore the dichotomy of ”see” and ”not see”(look, don’t look”) interactions with what we define as ”vision-based interactions” as an alternative to traditional eye-tracking gaze interaction for VR horror game elements*

...

*Our results suggest that we might be capable of expecting different interactions with differing intricacies to elicit stronger responses from users to VR horror game elements."*

Full PDF can be downloaded from:

llemmoo.netlify.app

# IT Product Development Design Project

## The DaSa Board

This project was a product the most extensive course during this bachelor's programme. We were tasked with creating a project that could fit within the frame of a given semesters chosen focus area. Our area became "Teaching informatics and computational thinking". The aim of this course was to see how we would utilize product development skills and techniques, in combination with the prior prototyping and software development courses.

The course emphasized thinking outside the box of easier interaction methods, such as touch screens, and move towards a multimodal approach that could not only receive complicated input, but also produce a comprehensible output. This was quite challenging, as it is much easier, not only for prototyping measures, to conceive of an idea where the input and output is through something like a touch screen or a terminal interface.

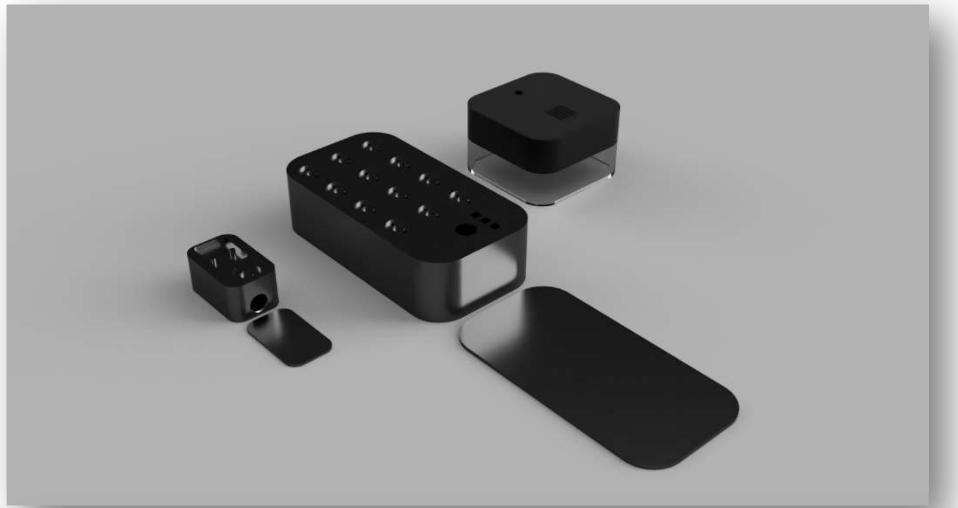
With the brainstorming skills we had honed through our studies, we sought to create a device, with a tangible user interface, that could teach the concept of data sanitization, by filtering sensor data, effectively through a little program, that the user would define using several "pegs" in a pegboard of sorts. This we called the DaSa board.

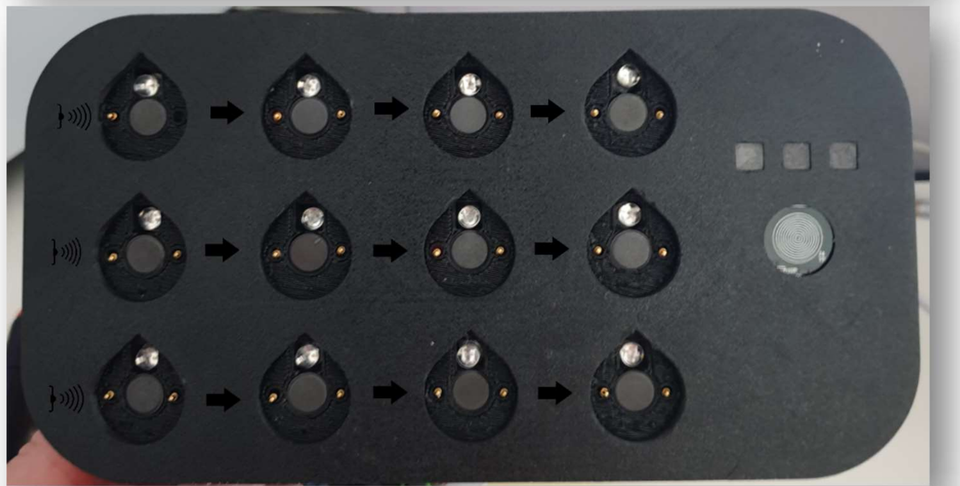
Our solution was capable of programming certain pre-specified outputs to react to user chosen sensor input. This input is programmable by the user as well, requiring the user to sanitise the sensor input to obtain meaningful control of the outputs. The tool has a button for selecting the output(three different possible outputs), as well as 12 slots to place "pegs" in. The tool comes with 12 "pegs". There are three sensor pegs, and three filter pegs for each of those sensors.

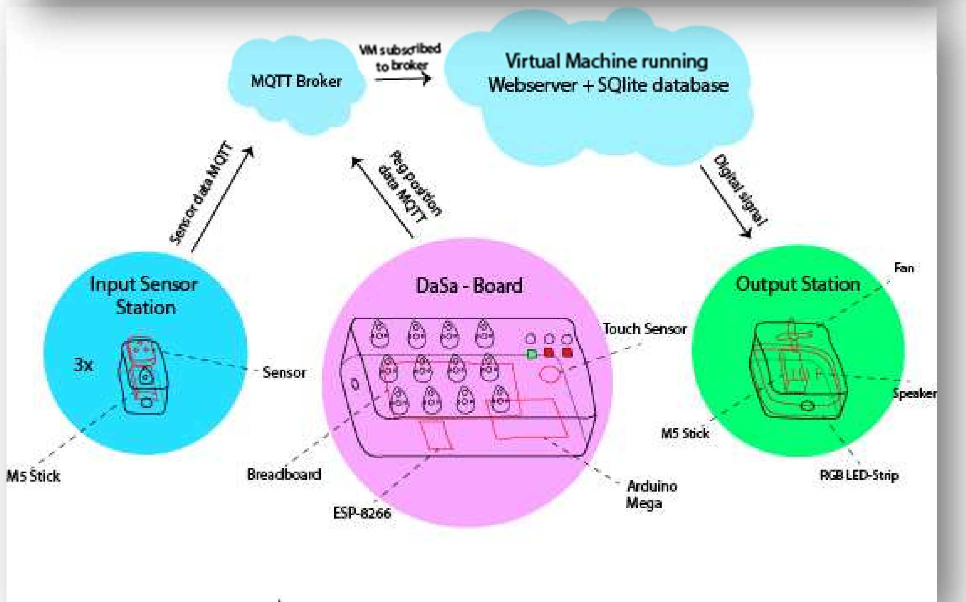
The course required us to make use of our understanding of business models(creating a business model canvas for said product), leverage our understanding of computational thinking and computational empowerment, make use of our experience building systems with IoT devices, as well as our extensive practice building and analyzing user interfaces and interaction designs.

During this project my main focuses was brainstorming and analyzing the solution, as well as building the physical prototype.

PDF of report can be made available on request







# Castle Environment, Unreal

**Learning to build materials, shape and build landscapes, utilize Quixel and MegaScans, as well as building environments with assets.**

Following the unfortunate change in pricing schemes for Unity, concerning both amateur developers and professional studios, it seemed that there would scarcely be a better time to learn a little about Unreal Engine.

Learning how to build this environment took efforts in how to make composite materials from texture, shaders, and maps, extensive work with the landscaping tools built into Unreal Engine, as well as changing and modifying assets in the world.

Building custom materials through blueprints and setting up lighting and other environmental properties.

Finding appropriate landscape materials through Quixel Bridge, to access the library of megascans, and understanding nanite vs LOD.

Finalizing the environment with shading, texture filtering, and landscaping properties, such as color, wind speed, wind sway etc. Background mountains and view distance done as well as clouds.

These screenshots are from said environment.

# Experimental Systems Development

## Tech-Wiki for Adservice

A tech-wiki built for Adservice as part of a course at AU. We were looking to delve into experimental systems development to acquaint ourselves with ethnographic methods and approaches to systems development.

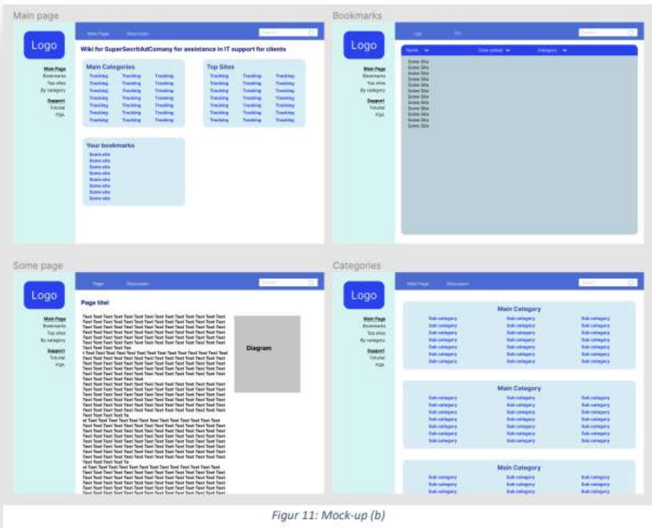
We ran exploratory interviews and workshops initially to discover issues and the best approach to solving those issues. We developed a prototype for which volunteers could provide feedback and subsequently re-developed into a more advanced prototype. The subsequent developmental phases were left at that due to time constraints, but the approach was meant to be agile. The results from this project were positive at the hands of the volunteers, and said volunteers were a part of the design process from the very beginning of the project, underlining the focus on participatory design and user evaluation techniques.

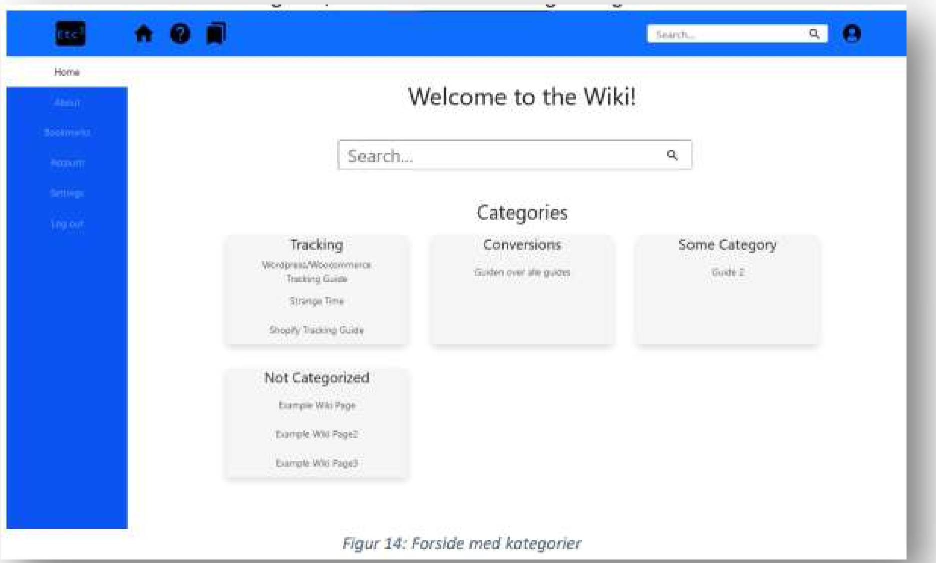
The mock-ups are done with Figma and the prototype itself is built with Javascript using the React Framework.

The UI is built with the MUI CSS framework.

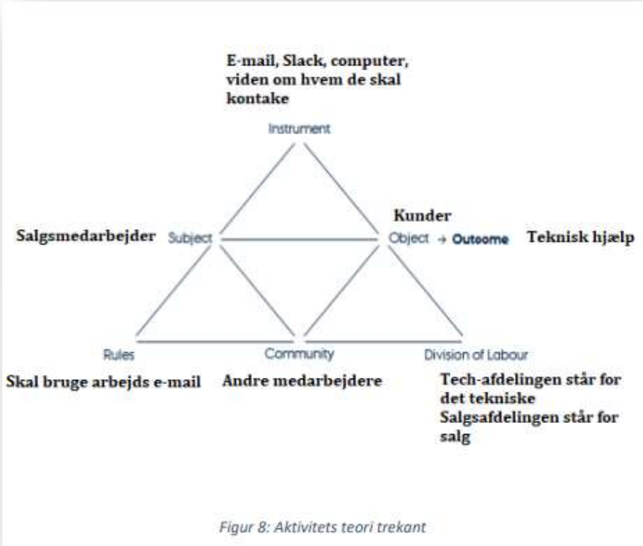
My personal contribution to this project ranged from building the project's web application, to writing and researching for the academic report. The first Mock-up was also done by me.

Figur 10: Mock-up (a)


Figur 11: Mock-up (b)



Figur 14: Forside med kategorier



Figur 8: Aktivitets teori trekant

Full PDF and GitLab link can be found here:

llemmoo.netlify.app

# Building the Internet of Things with P2P and Cloud

Sensor-data driven Web Application

When tasked to build a sensor data-driven application, we chose to provide a solution that would hit closer to home for us. We chose to work with a gaming-health monitoring platform for our solution. Provided with the sensors made available to us, we built something we could use in our everyday lives.

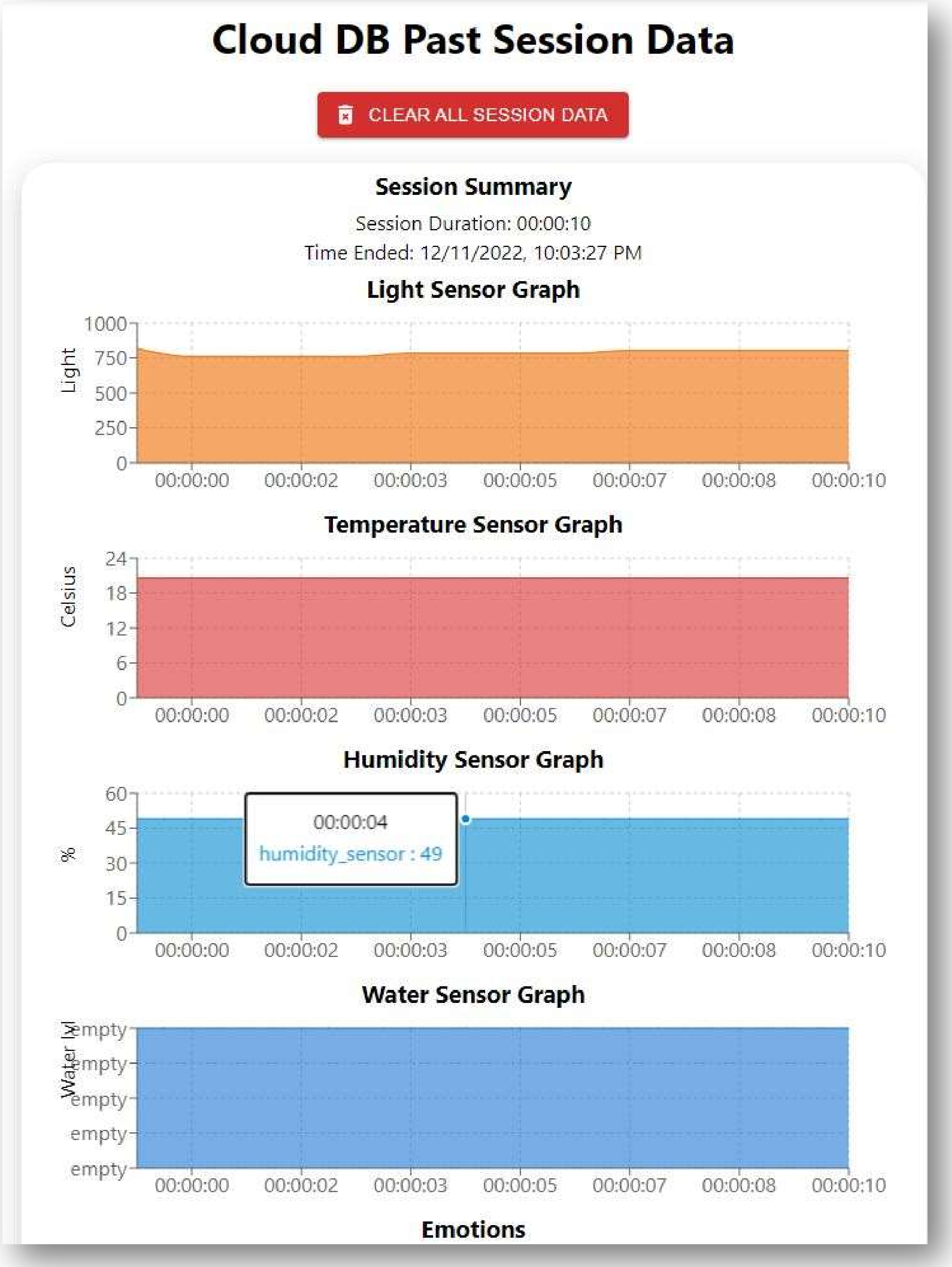
The web application was meant to be a way for the user to be able to track their gaming environment during sessions, and gain some data so that they could analyze their environment on the go. This was done through sensor data such as light, temperature, humidity and hydration(how much the user drank during sessions).

We also had a webcam tracking emotions for the user, prompting the user to calm down if they were upset.

We implemented the idea using ML, cloud technologies, and MQTT brokerage for our web application. For the sensory side of things, the M5Stack eco-system was a pre-requisite from the get-go and we decided to build upon this with an ESP32 Arduino to handle specific sensors. The M5Stack's ability to receive and send sensor data through it's MicroPython libraries made it easy for MQTT integration.

The web application was built with the React Javascript framework and the CSS framework MUI.

My personal contribution to this project was focused on programming of the M5Stack devices and writing and researching for the academic report.




Full PDF, GitLab link and Youtube Demonstration can be found here:

llemmoo.netlify.app

# The Web of Things

## IoT Course, M5Stack and Python

Course designed to build python backend development and the deployment of said backend on a Linux Apache server. The course was concerned with the development of IoT devices and coupling these devices to a front-end for users to utilize.

The course introduced us to the M5Stack eco-system of smaller IoT-able devices programmable with block-programming and micro-python. This course parted the scope of the course into smaller partial-assignments concluding with the development of a IoT device receiving sensor-data and transmitting this to the front-end of a web-application through the M5Stack's powerful MQTT library, and an MQTT server.

The final web application was built with a python backend coupled with native JavaScript front-end.

All efforts of this course are my own.

GitLab links can be found here:

llemmoo.netlify.app

# Software Engineering and Architecture

## Civilization Game

A course and project centered around the idea of real software development, instead of the less intensive and more prototype-oriented programming we had worked with up until that point. The project's intentions were to teach the valuable lessons of refactoring, agile development, and the benefits of producing reusable, refactored code.

This project taught me the use of design patterns in software development, and the importance of structured software development. The game was built using Java, structured in multiple assignments, meant to exercise refactorization and employment of design patterns in software as more and more advanced functionality was implemented.

We built a “copy” of the game Civilization and used this as the basis for our refactoring and extension codebase. The game would iteratively be expanded upon and refactored using software design principles and frameworks learnt through-out the course.

The course was my first introduction to the agile process and working collaboratively in teams on a single codebase, prompting one to learn lots about Git.

During this course we collaborated almost equally on the assignments and the reports were made as video demonstrations and walkthroughs of these iterative changes to our Civilization game.


GitLab link can be found here:

llemmoo.netlify.app

# Physical Prototyping

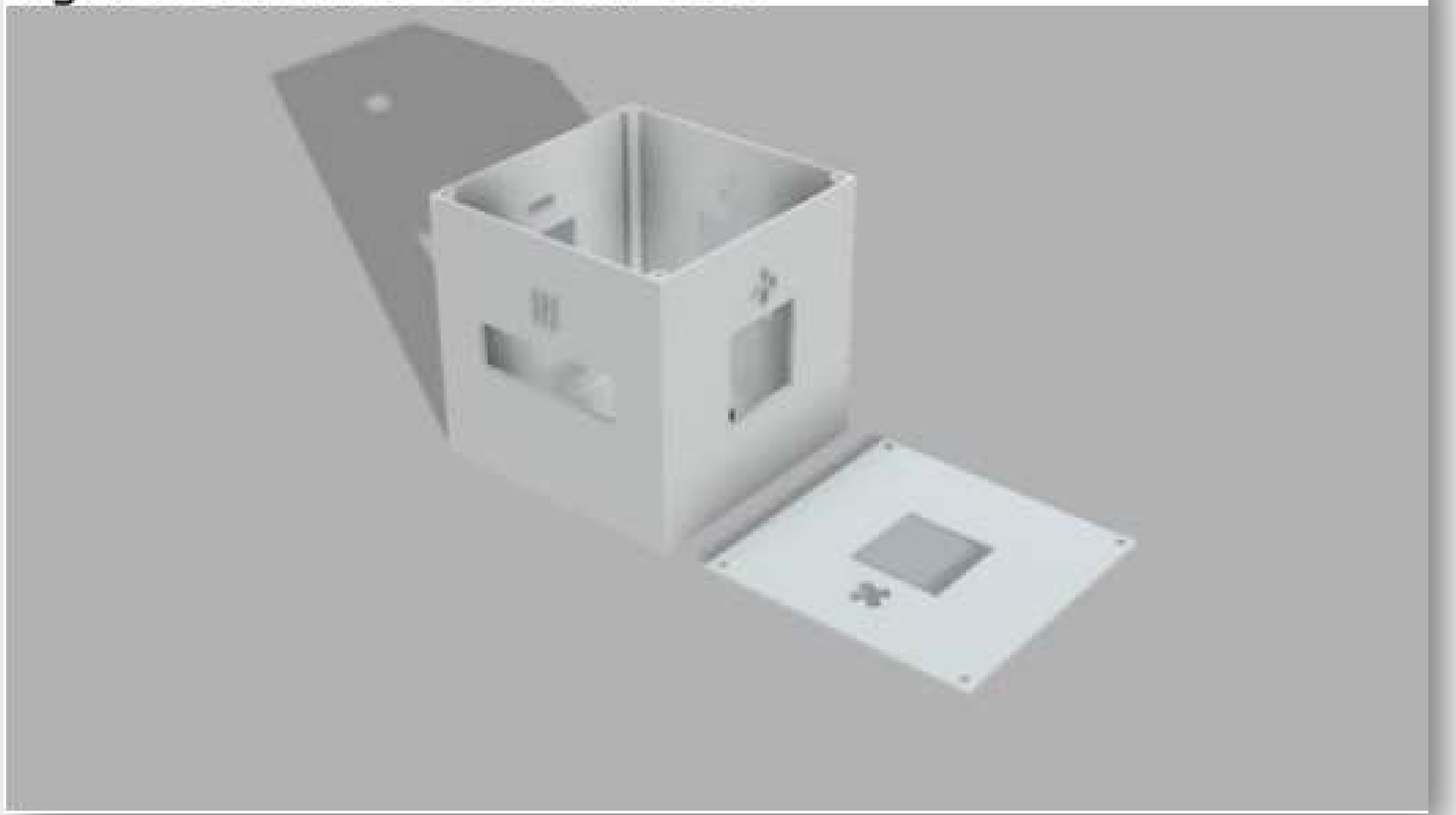
## Math Box

When following IT-Product Development, an integral part of the bachelor's programme is the capability for its students to be able to make physical, working, and capable prototypes. These prototypes need to demonstrate significant functionality of the ideas generated by the students, functioning as a proof of concept. As such, generalized knowledge of simple electronic components and circuits is needed, in order to create functional prototypes, especially for tangible user interfaces.

Having previously studied electronics engineering for a semester, I already had gone through entry-level electronics and circuits, which made me very valuable for the team when it came to building the more intricate circuits for the board.

We were tasked with building a physical prototype, that could support a specific learning goal for a certain age group. We chose to build a "math box" game, which could help young children train their arithmetic skills.

Figure 1: Render of Math Box



We built a box, with four sides that would each represent a specific mathematical operation. This would be addition, subtraction, division, and multiplication.

A screen would prompt the user with simple equations, with a missing operator, as such: 3 _ 3 = 9. The solution would be for the user to tap the side with the multiplication operator, where a force sensitive resistor pad is present, registering the users touch. The user would receive feedback of lights, sound and vibration depending on whether they gave the right or wrong answer.

This box was made possible by constructing a ESP32 circuit on an internal breadboard, controlling the functionality of the math box, and subsequently connecting this to several custom-made PCB's, made from scratch through acid etching techniques. Each operator necessitated its own custom PCB and circuit, requiring us to design an electrical circuit for each operator individually. These boards were all connected to a master board which ran the feedback and screen of the box.

During this project my main focus was designing and producing the custom-made PCBs present in the prototype.

PDF of report can be made available on request