# **IoT Experimental Project - DIG008AS3AE-3001**

Aku Liski  
Jonas Lemström

GitHub: https://github.com/Akkel-i/Data\_Duelists

# Introduction

We have done couple of IoT courses and we wanted to continue our project. The project is a NFC tag reader turned into a game. We got it to work with e-paper and everything, so proof-of-concept is working. Next step was to make it better.

# Ideas

We had the idea to improve our project that we started on the last course. It seemed that the esp32 computing power was not enough for us to implement some functions to the game. The idea was to send the data to a server and use computers superior processing power to tun our game.

# What did you need to learn

At least how to send data and the read it in the destination and how to use it to our liking.

Then when the plan changed we needed to learn all about Gamemaker program. It has inbuild functionalities that we needed to learn and it uses its own language, GML. So there was quite a bit of learning in store for us.

# The building phase

First, we got a piece of cloud from Haaga-Helia to send our data. That went surprisingly well, and we got text to appear in the server. But the problems started when we tried to read it and use it to anything. We tried to read the data for weeks even with the help of the IT-helpedesk, but to no success. There was some kind of reading blocking so we couldn’t access the data so a new plan was needed.

Then we thought how else we could use the NFC data to play a game and we ended in a Gamemaker program. First steps were again to connect esp32 to the program and send data. We got it to work with some third party imports, so that was all good.

Then we spend some time to make sense how to make the read NFC tag to appear in any sensible way in the Gamemaker program. That took some time and fiddling, but at the same time we worked with the game. We found a opensource extension to Gamemaker which made it translated esp32’s information to Gamemaker.

The progress was quite slow as it was a new program and language, so everything was a bit slowed than anticipated. We got working prototype done before deadline and we’re continuing the project to build a working game after the course.

# The outcome

Our first plan didn’t turn out to be a success, so we needed a new idea. We think that we got nice idea to integrate the esp32 to a real game engine, as it was the idea to make a game. We got the data from the esp32 to the game program and we got things to happen depending on the data, so we think that we did all right. Of course the code is ruff and game is barebone but as a proof-of-concept it works and the ground work is done.

We learned to persevere through problems and make do what we got.