Anup Jasani (momesmo@vt.edu)
Kenta Yoshimura (yoken@vt.edu)
Nhan Pham (pnhan@vt.edu)
John Stradling (jstrad@vt.edu)

# ECE 4564 Assignment 4 – Group 16

## Project Overview

Assignment four will utilize all four of our team's Raspberry Pi devices. One Raspberry Pi will be acting as the server and will be running the MineServer.py. The other three Raspberry Pi's will be running the client codes that will have a slight variation in order to check for a different token id. For communication between the client and server Raspberry Pi's, we will be utilizing AIOCOAP.

## Design Process

The team first started by working on their assignment parts of the project individually. Daniel was in charge of making sure the GPIO was working correctly with the LED and displaying the correct colors. He then made the LED change colors depending on which the current player's turn it was.

Anup was in charge of the COAP communication. He made sure that the server was running properly. Once the server was running, he built the communication aspects of the client codes to use the required methods to transfer the data. Once communication between the server and the clients was established, he made sure the the data was tupled and pickled correctly.

John and Kenta were in charge of working with the Minecraft API. They first made sure they were able to send commands from their python modules to the Minecraft API. Once they had the code working, they made the necessary calculations to build the wall and have the clients swap turns.

Finally the group combined their codes to have the client and server communicate with each other while building the wall and output the correct color on the LED depending on who's turn it was on the client end.

## Team Responsibilities

Our group decided to delegate responsibilities according to the three major portions of the Assignment. Anup was in charge of establishing the COAP communication. Daniel was in charge of the GPIO and using the LED to displaying the correct colors. John and Kenta were in charge of utilizing the Minecraft API and make the code build the wall. Although the team had delegated responsibilities, they would team up when someone ran into an issue in order to make sure everyone was on the same page.

## Conclusions

When first working with the Minecraft API, we had no issues because we were sending commands through the shell in a python version that was before 3.5. However we ran into the issue of our code not running in python 3.5 and only 3.4. We originally did not care however due to our implementation of COAP, we needed the code to work in python 3.5. In order to fix our problem, we had to copy the mcpi folder of the Minecraft API and put it into the repository of our Assignment 4. This was the mcpi module was in the same folder and we would run into no issues.

After fixing the mcpi issues, our group worked very efficiently to finish the assignment and everyone helped each other when needed. This project was very helpful as well due to the fact that we learned how to deal with updating python versions in order to utilize the COAP communication methods.