

CS 3340.502 Final Project

Benjamin Hickey

Social RGB Averaging Pipeline

When considering this project I was immediately struck by the idea of writing a web server in mips. At the time I didn't know if mips (via mars or not) had networking functions suitable for such a task. Upon finding out that mars is incapable of such a feat my thoughts immediately turned to other ways I might be able to abuse mars into some sort of api functionality. File io seemed like a reasonable solution so I got to work drafting a means of connecting mars to an external api. After experimenting with different io schemes I decided on cli-based io using pipes and syscall 8 to read strings from stdin.

With console input working I needed to decide exactly what input to parse or manipulate. I thought it would be really cool for the class to influence the project during the presentation, and looked for a way of doing this with minimal difficulty. After researching I decided to get input from a [twitch.tv](https://www.twitch.tv) chat room and feed to the application to manipulate as it needed. This works well because the twitch api is very well-documented with plenty of examples to draw from. Additionally, twitch chat rooms are essentially just irc rooms with a pretty mask on, so there are plenty of libraries available for connecting to them. As previously stated mips does not include any networking features baked in, so I had to write a small node script to take the messages

from the chat and send them to my app. Getting the messages and printing them to the console was fairly simple, and pipes take care of the rest of the input into mars.

With input from twitch working I needed to decide exactly what to do with this absurd little presentation. I asked a few friends what they would want to do with a twitch bot during an in class presentation, but couldn't reach a conclusion. During this time I worked to develop a side-project using a raspberry pi and a strand of rgb LEDs. Naturally, I daydreamed about combining the two and how it would work to have the class control a string of LEDs during my presentation. After testing the limitations of the raspberry pi as far as networking, I realized that this was in fact possible. Having worked with rgb values in that side project, I decided it would be interesting to have the class collaboratively build a color from rgb values based on frequency and see it change in the LED strip, sort of a social experiment to see what color would be most popular.

The bulk of this project proved to be averaging the values from twitch chat using mips. The application keeps a tally of how many messages are "red", "green", or "blue", as well as the total. After each message it uses these values to calculate the relative rgb values from 0-255 that each color should have based on its frequency. These rgb values are then printed to the console. Unfortunately this is the extent to which the project can be tested without hardware, the next part (the hardware control for the presentation) requires a number of materials that I can't expect peer reviewers to have on hand.

As for the LED strip, it's actually a matter of 10 lines of python to parse the rgb values and apply them to the LED strip. The interesting part proved to be implementing

the pipe between mips and python in order to make everything run smoothly, as well as ensuring that all the necessary software was installed and up to date across multiple systems for testing. That said, watching twitch chat control an object in the real world is really really cool and I look forward to seeing it live in front of the class.

Installation Instructions

I have tested this project on Windows and Linux, though theoretically it should work on MacOS if the proper package management systems are installed.

The core pieces of the project (chatfetch and main.asm) require:

- The source folder for the project, including mars.jar
 - Either unzipping the folder or running “git clone <https://github.com/bensometime/archfinal>” and “cd archfinal” should be the same result
- Nodejs ~8 to be installed on the system
- Npm (node package manager) to be installed on the system
- Tmi.js to be installed on the system in the source folder (this is a library used for connecting to irc servers and makes getting the messages from twitch chat very simple)
 - This package can be acquired by running “npm install” in the root of the source folder with the previous requirements satisfied

Additionally, controlling the LEDs used in the presentation requires:

- A raspberry pi
- A breadboard
- Some wires
- A NeoPixel LED strip
- The adafruit-circuitpython library to control the lights

Running the Project

With the dependencies satisfied running the project is fairly simple. I've made a couple of launch scripts that can be found in the source folder. Start.bat and start.sh should work to launch chatfetch connected to the mars app, with resulting rgb values printed in the console. An important note: testing this will require internet access and a twitch account in order to send messages to the chat at twitch.tv/botnamedassembly. Typing the messages "red", "green", or "blue" into chat should result in adjusted rgb values being printed to the console.