

CDEFO Week 14 (#13)

Timeline:

Week 8 - End:

- Build user experiences, curate content, and test stability. (until the end)
 - Write any remaining C++ functions
 - Make the experiences pretty and bug free
- Building the user experiences is probably going to take the longest, as it is going to have some QA involved to make sure they are actually enjoyable. Lots of talking to users.
- Write documentation for the C++ library (until the end)
- Prepare for demonstrations

So, as of writing this, I have all of the frames ready to be assembled, all of the curtains cut and hemmed, all of the software cleaned and GUI'd up, and the musical experience that I plan to debut. After wrestling for weeks with LED bug that turned out to be just a loose pad under hot glue, I finally have all 5 LED strips working flawlessly. In terms of making everything bug free, I found a bug that had been affecting me for a while since it messed with the seamlessness of the whole experience. This was the case when the Arduino would the presence of an NFCtag when there wasn't one on the receiver, and it would hang up the system. This was an issue since I needed the experiences to finish even without a tag present, and short of precisely timing the experiences, it is more useful to just check and see if the tag is present, and cut the experience off at a reasonable time (i.e. between lighting scripts). For the longest time, I had just assumed it was a memory issue, considering how large the Adafruit_NeoPixel objects are relative to the amount of RAM in the Arduino (one object can be 300 bytes relative to the 2000 total bytes available). The, however, was not the case; issue here was that the built-in function to check the NFCtag presence was implemented using functionality that isn't available to the Arduino compiler, but is available to most other C++ compilers. That is, the "stdint.h" library (which is 3 layers of abstraction away from the NDEF library) has a function called "readResponse", which reads from the "standard input" and defaults to a timeout = 0 when no timeout is specified in the function call.

```
virtual int16_t readResponse(uint8_t buf[], uint8_t len, uint16_t timeout = 1000) = 0;
```

The = 0 in this code snippet represents the default timeout of the default stdint.h library. The issue is, though, that the Arduino compiler uses a different version of this file, which has a different timeout (I believe around 1 second). As a result, whenever I'd call the function without a tag present, the entire program would hang for an entire second before doing anything. As a result, the program would hang and stutter an appalling amount. In this instance, calling a non-default, sufficiently small value for the timeout (which is measured in milliseconds), such as "NFCTagPresent(10)" works perfectly; however, "NFCTagPresent(0)" and "NFCTagPresent()" stutter. Now, you'd think that "NFCTagPresent(0)" since it ideally would force the timeout to 0, but it actually doesn't, because when this top level function reads "0" it sets the default timeout, which isn't 0.

```
boolean NfcAdapter::tagPresent(unsigned long timeout)
{
    uint8_t success;
    uidLength = 0;

    if (timeout == 0)
    {
        success = shield->readPassiveTargetID(PN532_MIFARE_ISO14443A, uid, (uint8_t*)&uidLength);
    }
    else
    {
        success = shield->readPassiveTargetID(PN532_MIFARE_ISO14443A, uid, (uint8_t*)&uidLength, timeout);
    }
    return success;
}
```

You can see that the function call within the “timeout==0” if statement has no timeout specified in the readPassiveTarget function call (which is an abstraction of readResponse). So it sets the timeout to default, which isn't 0.

Now that this bug has been fixed, the experiences move as smooth as butter, with the proper transitions and timings without and hint of glitching or memory leakage. Also, I finished writing the GUI in Python using a default library called Tkinter, which is simple widget-based GUI, which is really all I need. The most important part is that I give it a non-offending color palette and a good layout, and use it sparingly enough that it doesn't detract from the other audiovisual stimuli I have planned.

Tuesday evening I'm going to get someone to help me bring down all of the speakers and the frames so that I can do some last minute inventory and some hooking up. I'll pretty much just have to hope that the curtains and the frames will setup well, since this will be the first time I have everything setup.

I ended up working on this for around 8 hours this week, not including the time I'll spend Tuesday and Wednesday building.