**How to use the coded files**

**Demo:** <https://youtu.be/DcfJYrF_yjs>

Initially run mininet with the custom topology using command line:

*sudo mn --custom ~/sandbox/jtopo.py --topo mytopo*

-replace *~/sandbox/* with the files current directory

Make sure Xming is running. Xming allows mininet to open separate terminals for each host using:

*xterm h1 h2 h3*

-in our case, we have three hosts

After on each terminal, launch the programs in order: Server--> Client--> Renderer. It’s important to note that our program was created using Python 3 so make sure to run them using *python3*

Then use Controller to show commands with *!help*. From there, simply make requests from the Controller.

Note:

* Run programs using *python3*
* To clarify ‘filename’ does not include the quotations, you just need to put filename name with the extension.
* Server only supports steaming of text files (extension .txt).
* Commands resume, pause, and playback only work after a file is streaming.
* Commands are case sensitive.
* Controller is the only host that takes user input request, hence the prompt symbol.
* ls-local prints the list in the proper format. ls-remote prints the list in an incorrect way. The issue happens when the controller receives the message, either from encoding or decoding.
* The program uses “10.0.0.1” as the server address when launched on mininet, but can be changed to “localhost” when running off mininet.

**Design**

When we were in our first phases, we decided to keep the implementation close as possible to the requirements. In which, we did. The server, controller, and renderer work exactly how it was explained in the scope of the project, and at minimum the text files were supported.

Initially we were excited to go beyond and implement video transfer. The reasons why we couldn’t implement video is because we would have to use a module called “cv2” in python. It creates a GUI for video playing and creates stream for video playback. While it would be neat to implement that into our code, we had to cut that part out because the group working on the project only consist of two people. Also, other reasons for cutting “cv2” is because that means if the TA had to run the program he would need to have “cv2” in Mininet to run it on the node(s). So, we took that into account also.

**How the Topology looks**

Switch