ADVISOR: ANDY SOMOGYI
STUDENT: WILLIAM CUTCHIN

JUPYTER NOTEBOOK SIMULATION WIDGET

FALL 2020 LUDDY DEMO WEEK

INTRODUCTION

In this project I worked on creating and understanding a new widget within the Jupyter Notebook environment. This widget would allow for any data, event, or video to be streamed through a series of images. This directly solves a problem that Professor Somogyi faces with his simulations. We thought that this widget would find many uses in comparison to the video widget that only accepts. mp4 files.

UTILITY OF THE WIDGET

This widget, when complete, will be able to take a stream of still images and allow the user to stream them into their jupyter notebook and display the live images. This has many more applications than the native video widget since there is no throttling due to file type. Eventually we want to incorporate options for interact-ability, being able to manipulate 3-D planes to look at models or simulations.

CONSTRUCTION

In the mission to create the widget, we found that there was a natural evolution to get to the point where we can play a video-like stream to the widget that we can then manipulate. Starting with the image widget displaying a list of images, then using a slider to run through the individual images, then setting their change to a timere where we can essentially set a frame rate, then add large biaxial sliders to allow the manipulation of the images angle in the future.

IMAGE WIDGET

```
[1]: #import modules

     import glob
     import matplotlib.image as mpimg
     import matplotlib.pyplot as plt
     import ipywidgets as widgets

     #global variable initialization

     path = "/Users/willcutchin/Desktop/jupyter-project/jupyter-images/*"
     imgLst = glob.glob(path)

[2]: #img widget

     x = 0

     while (x <= 14):
         file = open(str(imgLst[x]), "rb")
         image = file.read()
         w = widgets.Image(
             value = image,
             format = "jpg" or "png",
             width = 600,
             height = 600,
         )
         display(w)
         x += 1
```

SLIDER WIDGET

```
t modules

 glob
 asyncio
 time
 random
 matplotlib.image as mpimg
 matplotlib.pyplot as plt
 ipywidgets as widgets

l variable initialization

 "/Users/willcutchin/Desktop/jupyter-project/jupyter-images/*"
 = glob.glob(path)

r widget initialization

dgets.IntSlider(
n = 0,
x = 14,
ep = 1,
scription = 'Image:',
sabled = False,
ntinuous_update = True,
ientation = 'horizontal',
adout = True,
adout_format = 'd'

r output

plot(z):
= z
le = open(str(imgLst[y]), "rb")
age = file.read()
= widgets.Image(
lue = image,
  format = "jpg" or "png",
  width = 600,
  height = 600,

splay(w)

s.interact(replot, z = widgets.IntSlider(min=0, max=14, step=1,

z  0
```

EVENT THREADING

```
import asyncio
import time
import random
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
import ipywidgets as widgets

#global variable initialization

path = "/Users/willcutchin/Desktop/jupyter-project/jupyter-images/*"
imgLst = glob.glob(path)

2]: #slider widget initialization

z = widgets.IntSlider(
    min = 0,
    max = 14,
    step = 1,
    description = 'Image:',
    disabled = False,
    continuous_update = True,
    orientation = 'horizontal',
    readout = True,
    readout_format = 'd'
)

3]: #slider output

def replot(z):
    y = z
    file = open(str(imgLst[y]), "rb")
    image = file.read()
    w = widgets.Image(
    value = image,
        format = "jpg" or "png",
        width = 600,
        height = 600,
    )
    display(w)

widgets.interact(replot, z = widgets.IntSlider(min=0, max=14, step=1, value=0))
```

z 6

FUTURE STEPS

The next steps are connecting the client side (Jupyter Notebook) with whatever server side application that is being used to run the simulation or desired output. This process will take time due to the need to understand how and when these interactions take place. Beyond this, adding interactability with the widget would further improve its case use and appeal.