Week 3

Completed on November 11

We have decided to use GitHub as the primary source of sharing code. GitHub allows users to use any programming language and have it placed into GitHub for others to use. GitHub is great for group projects that can be used to share code. Instead of sending the .java or .m files or sending a word document (very tedious!) we can just go to GitHub and copy and paste the code right way into our programming language that we a are working on. This way it will not only save time but also allow professor Martin to look at our code. Each time the code is updated, we can get notifications that the code has been altered. Also, some features that I would like to include are 1) the ability to create files. Most of the times one project is not enough. People may have multiple projects they are working on. GitHub allows you to create new files, which is another reason why it is better to have GitHub. 2) If files have been updated and say you don't want people to get confused on which is the newest file, you have the option to delete the code and post the updated one in the same folder. 3) You can have other people comment on your code.

Completed on November 12

Professor Martin wanted us to have a line variation that assorted colors. What we originally had was a line variation that showed the proportions. The highest occurrence showed at the bottom. The lowest showed up at the top. After we met with Professor Martin, she wanted us to have it so not only is every color at the same spot (if white is on the 8th spot the first frame, it will stay at the 8th spot for every single frame after that), but also have it so that the line shows up horizontally rather than vertically. She got this idea from the app created by Theodore Gray. In his app, he has it so it shows the line color variation being updated horizontally in a straight line. This makes it cleaner because the colors are not all over the place. If you see white at the bottom then you should see white at the bottom again. Otherwise, it will look messy.

Completed on November 13

We created a 3D plot quite early than expected. Trush tried to create a 3D plot but was unsuccessful. Yao created a bar graph with the help of Trush and Shane. He was able to do the bulk of the coding, but Trush and Shane helped fix up some errors encountered in the figure. After spending 2 hours or so, Shane was able to get a 3D figure to show up and have it presented before the week ended. We basically finished our project just this week. This was what we had in plan to be our finished product, but we are moving so fast that we honestly feel like this was done weeks ago when it was done just this week. Right now we completely finished data analysis of an image, and data analysis of a video. We were able to get our program look very similar to the project that Theodore Grey created. We truly enjoy working together, we all have the dedication to make this project go really far. Not only did we get a 3D figure to show up for the image, but we also got a 3D figure to update the color proportions every frame of the video.

Here is the link to see the 3D figure update for a video! Link: <https://www.youtube.com/watch?v=2DLCQYCy_aw>

Completed on November 14

Here is what Trush found in the API's that we can use for MATLAB. The first one gives a detailed explanation of how we can use this specific API.

MATLAB API

.dom.\* api: You can use this API to create reports. Reports such as Word, PDF, HTML. This way a user and access code written outside of their home. This can be accessible outside of MATLAB. Here is an example of how the MATLAB .dom api works:

1) Import the package .dom.\* next below this code you will have to type rpt\_type to specify the type of report you want. If you want a pdf you type it in single quotation marks. If you want word it would look like ‘Word’

Import mlreportgen.dom.\*

rpt\_type = ‘pdf’;

2) Next you create a variable name. For example I have made the variable name called doc. After the equal sign, it is specified as document. This does **NOT** mean It is going to be a word document. It is basically going to create a empty pdf document. The document will be called mydoc. The append function will display the next. In this case Hello World.

doc = Document(‘mydoc’, rpt\_type);

append doc, ‘Hello World’);

3) The close method will close the variable and document name doc. The rptview(doc.OutPath); will open the document in in-built PDF viewer.

close(doc);

rptview(doc.OutputPath);

4) To create a paragraph, start off by naming the variable. The variable name for this code below is paraObj. Next to the equal sign is Paragraph. This will create the paragraph. Inside the parameters, the message gets displayed. Append(doc,paraobj); basically will append the paragraph made to the document

paraObj = Paragraph(‘This is a paragraph’);

append(doc,paraObj);

5) To insert an image use the code below. First create the variable name imageObj. After the equal sign write Image to have it display an image. The which part is to specify which image you want to use. For our example we are using image image123. Below this code we specify the width. The width we have set is 1.52 inches. This will be the with in pixels. Below the width, we have the height specified. The height will be 1 inch total. The append(doc,imageObj)); will add the image to the report.

imageObj = Image(which(‘image123.jpg’));

imageObj.Width = ‘1.52in”;

imageObj.Height = ‘1in’;

append(doc,imageObj));

6) The code below will be used for creating tables. First name the variable like always, next write Table to specify that it is a table. Inside the parameter allows you to create a 6 by 6 table. Append(doc, tableObj)); will create the table into the document.

tableObj = Table(magic(6));

append(doc, tableObj));

Some other useful API’s:

<https://www.mathworks.com/videos/upgrade-advisor-api-121582.html>-This is video shows you the demonstration. To summarize what I learned from this video is that it allows you to update the models. It will not only fix the models, but also show recommendations on what should be updated. So for Shane and I, we encountered an error that didn’t allow us to see the pixels displayed on the 3D graph. What this API allows you to do is to basically prevent that from reoccurring. It will analyze things such as the frame, model, functions, and properties. This will prevent us from having another problem to occur the day of the meeting with Professor Martin!

<https://www.mathworks.com/videos/faster-simulations-with-performance-advisor-90544.html> This allows you run models much faster. If the program finds any sort of way to speed things up it will do it. However, I have only seen logic gates being used. Couldn’t find any information on whether it can work on our 3D model.