Trush Patel- Documenter

Yao Sedzro- Management

Shane Staret- Lead Programmer

Every Week Detailed

Week 1

Completed on October 27

We plan on meeting as much as possible. However, we have certain times that we can meet up. Here are the times that we are usually available.

Shane:

Monday/Wednesday/Friday: He is available for the most part. Has a class at 1pm and then has one later on the day, but that’s not until 6pm.

Tuesday/Thursday: Before 12:45pm

Trush:

Monday: 10am-1pm

Tuesday/Thursday: 11:30-2pm

Wednesday/Friday: 10am-3pm

Yao:

Yao, for the most part, stays at Montco very late.

Monday/Wednesday: 12pm- Whenever he feels like going home

Tuesday/Thursday: 11:30-2pm & 4pm-Whenever he feels like going home

Friday: 10am-Whenever he feels like going home

Completed on October 29

We considered source codes to help us with MATLAB. We only have so much time to complete this project. Therefore, we need to know every single code and fast. We must spend most of our time trying to figure out how the syntax works. After that, we must learn how to code in MATLAB. Therefore, we considered open source codes. The link that I have found allows you to look at a wide variety of source codes that we can use for our project. In the link that I provide, you can see every single code that you will ever encounter in your life when programming with MATLAB. The good thing is that not only will it show you every single source code in the MATLAB programming language, but it will also show you the proper way to use the code. So for example, if you wanted to read an image, you would do instead. This link that I provided will show you not only the way you're supposed to use it, but it will also

Completed on October 31

Today we found links to image manipulation. It was very detailed and can be very useful. This one link we found: <https://www.google.com/url?q=https%3A%2F%2Fprocessing.org%3A8443%2Ftutorials%2Fpixels%2F&sa=D&sntz=1&usg=AFQjCNE54zfW7XzSV5zeYzoObC8WtFOh0Q>         has everything we can possibly need. It has the code which we can refer to when working on this project. We made good progress and almost began to start coding if it wasn't for our tight schedule. The link that we found can help us with color saturation (one of the main topics from that Disney Animation article that Professor Martin sent us. Also, we looked at some courses because we plan on using a program called MATLAB that is used primarily for image processing. Although we had to spend some out of our pockets, it was well worth the investment and can make our project much more reasonable with the resources this Udemy course provides.

show you common errors that people mostly make. Also one more thing I would like to mention. This link provides you with codes that you can play around with for almost any topic! Here is the link: <http://people.sc.fsu.edu/~jburkardt/m_src/m_src.html>

Week 2

Completed November 5

Today was our second time we met up. We discussed our ideas the first meeting. Second meeting was with Professor Martin, we all pitched our ideas and she gave feedback on how we can expand our ideas. Shane's idea was to create a way for a user to take pictures and see the elements. Yao's idea was to create an alternate source of energy and use Arduinos to operate. My idea was to create a drag and drop app that allows you to code with ease. The idea is sort of like the game designing application Alice. Today we narrowed it down to one project, and created deadlines for the future and some meeting dates.

Completed on November 7

Nobody in our group had experience with MATLAB. We needed a good foundation to start programming. We looked through numerous courses to help build up the foundation. After looking through a couple of courses on Udemy, we found one that attracted us. Just from looking at the course description and the content, it looked like it was very useful for our project. Unfortunately, only the first section came to use, the rest of the sections were primarily used for editing pictures. We were hoping that this course would show us how to plot data, create things such as a 3D graph, and help us for our idea on using our phone or any kind of camera to do certain things. We made sure that we finished the entire course before moving on. Trying to collect as much information. We ended up having to request a refund because even after we finished the entire course it did not come to use at all. We are currently looking for more courses out there on Udemy because we believe that Udemy offers lessons that are much more in-depth than you would find on YouTube.

Completed on November 7

Our code is currently being updated. We met with professor Martin, and she wanted us to make the line graph neater and have it so that the colors are in the same spot. So if white was on the bottom the first frame, we WANT it to be on the bottom the next frame too. It doesn't matter if it is not the most dominant color. We must make it, so the line is neater. Also, we would like to include some more colors. We are currently working on adding more colors to the project. We originally kept it to six colors (Red, Green, Blue, Grey, White, and Black), but we got feedback from professor Martin saying that we should include more colors into the figures. What we have in plan is to include 20+ new colors. Also making sure that we cover all the colors of the rainbow. ROY G BIV ( Red, Orange, Yellow, Green, Blue, Indigo, and Violet). Then we will add some extra colors. These extra colors will, for the most part, be different shades of different colors. This is just used to add more varieties of color, so the figures can be more pleasing to look at and show more detail than just six colors. Also, we are going to create a new MATLAB folder just for the video version. Everything that we made for the image will be used for the video. Same figures, except the data, will alter, and new line graphs will constantly be added. The line graphs will connect with one another instead of showing a gap in between them. I have attached a list of colors we hope to incorporate into our project. This was created by Yao https://freedcampfilestorage.s3.amazonaws.com/A\_Project\_IzH/Color%20Variations%20MATLABs-39201.pdf?AWSAccessKeyId=AKIAI56FRGV4ETJQCNMQ&Expires=1512358577&Signature=slzu4JWqpz32DFaR8syV4%2B5SQTk%3D

Completed on November 8

We continued to watch MATLAB videos on Udemy that were for free. However, we wanted a course that was longer than just an hour and a half. Therefore, we used YouTube. We thought we wouldn't find anything useful because there isn't a way to build up a foundation. We would constantly have to watch videos that only have to do with things that we want. In other words, no video is going to completely give us the results we want. We must gain some knowledge first and apply that knowledge to alter some code that we watch from YouTube to have it output what we imagine it. MATLAB is not as popular as other programming languages, but it's good enough to have some videos out there to use. MATLAB is easy to adapt to and the features it must make it such a wonderful programming language to learn. We can create bar graphs, manipulate images, create 3D plots, do some interesting things such as have it detect your fingers, or face. It can be used to help create a more secure system as it can be used as an alternative way to lock a device.

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 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 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 .doom 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 an 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.

Week 4

Completed November 18

Trush had the video manipulated since week 1, and had the video displayed on the figure. All we had to do now was to add the data analysis part to the program. Shane was able to get the original video to show up on the top left and the manipulated video to show up on the right. Below both video is a bar graph that shows the color proportions for every frame until the video was over. Always it calculated the differences live! Below the bar graphs, we had the 3D cluster to show up. It would show the 3D cluster being updated every frame for the original video and manipulated video. Also we included the color line variation to have it look like the Disney Animated app created by Theodore Grey. We struggled with this, but the man himself was able to help us create it! Here are the links to the Manipulated video Data Analysis: <https://www.youtube.com/watch?v=2DLCQYCy_aw> <https://www.youtube.com/watch?v=oD7uPBSo3xM>

Completed November 18

Shane worked very hard on this. This was possibly the hardest thing he encountered while coding. He flew by the other parts of the project easily, but this was slowing him down a bit. Therefore we contacted Theodore Grey. Theodore Grey was kind enough to figure us part of his code to help us create the color line variation. Originally we only had 6 colors, but after adding more colors, we ended up with a total of 30+! Now, the color line variation looks way better than it did before. With the 6 colors, it looks very low, quality wise. When we added more colors, it looked more modern. We hope to carry this project one day to get our color line variation to show up just like how Theodore Grey had his color line variation to show up. Here is what Shane has made so far! Pretty cool!

Here is what we had before:

<https://www.youtube.com/watch?v=2DLCQYCy_aw>

Here is what it looks like now:

<https://www.youtube.com/watch?v=oD7uPBSo3xM>

Completed on November 21

This week, we have gotten a lot done. We researched how to get more colors into our program. We want more colors, so customers can see a more detailed analysis. The reason we I say, customers, is because we are trying to sell this idea, not just created because it's a project that we will get graded on. If we treated like this project was to get a good grade on the final then we would have low expectations. If we treated it like we were selling this, then we would be able to put more effort. So to make it a product worth selling, we included more colors to get a more detailed analysis. So far what we did was, we researched the RGB values of different colors, then we looked at the proportions of each RGB value to get it to display specific colors. Yao did a very nice job while researching the different color values. All we had to do essentially was to find more colors to add and then use the proportions. Wasn't too difficult, but was time-consuming as we had to make sure the values were correct.

Completed November 21

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 an 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.

Completed November 21

Shane was kind enough to explain his code. Since he is the lead programmer, most of the code that we have comes from him. He played a big factor in getting this project where it is right now. Trush helped him document some of his code, but Shane did most of the code commenting since he is the one who wrote much of the code. He knows his code better than any of us. Therefore, to help us understand his thought process when he was creating all his code, he made a video to help explain what was going on. He made an hour long video right after he came back from work at night. He was able to go over almost every single line of code with detail. And was totally honest when he said that even he himself didn't know what every code does. Shane explained everything thoroughly for both Yao and me to know what is going on. Both of us feel more confident with coding in MATLAB now after he made the video for us explaining the code and watching it.

Here is the link to his code explanation:

https://www.youtube.com/watch?v=NfeG2EDdjmE

Completed on November 21

This part was very cumbersome. Converting in MATLAB code to JAVA code was impossible. Literally. MATLAB can only convert to either C++ or C. Unfortunately, we couldn't do this either if we wanted to. We would have to buy a subscription to do this. Trush met with professor Martin on Wednesday to have a chat with her about this, since he was assigned to this part. She mentioned how if I couldn't get the code to convert from MATLAB to JAVA then I could at least make it, so JAVA can run MATLAB code to have it display the figures. However, even after , we had to make sure the school offered the JAVA Package SDK otherwise, this part would also be impossible. Trush contacted MATLAB through email and called them up. Both said it was impossible to convert the code, but it was possible to run MATLAB code on JAVA. The email provided the steps to take to have it do that. Trush doesn't feel comfortable with JAVA, and Shane didn't feel comfortable with Arduino. So they switched roles. Shane was able to get MATLAB output of the color proportions saved as a text file and then used that to create bar graphs on JAVA. Trush, on the other hand, took responsibility for getting the Arduino servo to spin in different directions, and have it display a "This is (Color)".

Completed on November 21

What we have in mind for this Arduino project, is to first create a wheel. Now, we can’t have all 30+ colors onto a wheel because this servo is about the size of half of a thumb. Therefore we have limited the color wheel down to just 10 colors. Red, Orange, Yellow, Green, Blue, Pink, Purple, Black, White, and Grey. We also must make it, so the servo pointer is bigger. Right now the pointer is as big as a USB port. It will be hard to tell which color it lands on as the colors are already crammed together and the servo can only rotate one hundred eighty degrees. Next, we plan on making it so the servo spins at different angles. Trush was able to do this and have it say which color the needle lands on. He did this while swapping positions with Shane as Shane was working on getting MATLAB code to run on JAVA. Now, all we need to do is have the color wheel made and have it, so Arduino can retrieve the color proportions that JAVA got from the text file it got from MATLAB's color proportion output. We will create an Arraylist that will give the data of the color proportions to Arduino.

Week 5&6

Completed on Dec 3, 2017

This week we hope to incorporate one of Trush's API's. He found three APIs so far. 1) Using .dom\* to create Word, PDF, and HTML documents. When we usually send codes, most people who are just starting off, send codes through Word documents. This can be a bad habit as its inefficient, but for beginners, it can be easy to work with. Also, another good use of this is that you can see your code straight away rather than having it sent to a word document then have it sent to a pdf, so you could see it on a phone. 2) We found an API that will fix the code for you. By fix we don't mean like fix the code so you don't get any errors, but rather make it so if the code ever gets updated, you don't have to go back to make sure that it works or not. It would automatically fix it for you. This is one problem that Shane and Trush faced. Both got an error saying that the version is outdated and that some features have been disabled. What ended up happening is, we couldn’t get the 3D cluster to work. It would show the graph, but it wouldn't plot any of the pixels proportions. 3) The last API Trush found was a way to speed up the output. Our video takes a very long time to run. This is mostly the case when we have an HD video. The reason why 3D videos take longer is that there are more variety of colors in HD videos. Therefore, the color line plot moves very slowly. Another thing is that we tend to skip a lot of frames to speed up the process. We shouldn't need to do that if we want to make this a project that can be built on in the future. This API could relieve a lot of tensions, if not all then at least somewhat.

Completed on Dec 3, 2017

We hope to add descriptions to MATLAB classes so it's easier for people to understand what is going on with our code. Shane did a very good job going into detail of almost what every part of the MATLAB code does. He made a video about it too. Here is the link: <https://www.youtube.com/watch?v=NfeG2EDdjmE&index=3&list=UUmQA16swmtPa29pRo9YtRTA>. Although, this doesn't include our finished product of MATLAB (still needed to add more colors to this, after the video was made) it explains everything that we have done so far. This may be helpful to those who really want to get into what we are doing, but the thing that really makes a project more interesting is a very short description. For example, people on stack don't want to see all your code. The longer the code the faster people close out of your question. They get discouraged to help because the code is too long, and is mostly code. That's why we have it, so our code is shortened, and briefly explains what is going on.

Completed on Dec 9, 2017

We are currently reformatting the display of MATLAB code. We want to make sure the code is condensed so it does what it is supposed to do, and not show a bunch of code. Also, as usual, make sure things such as indentation is correct. Nobody wants to see code that is not properly indented, (especially on stack!). We want to make the code look appealing to the user, not a pile of junk. We have a ton of code and the last thing we would want to happen is have some code misaligned to the rest of the code. Luckily, programming languages make it easier for us to do it all in one step. Next what we need to do is create a way to spin a 3D plot. We most likely won't have this done because we are working with Java and Arduino to get another finished product, but it's a cool feature we can look  into if time permits. Another thing that we had to do, and got completed is changed the dots in the 3D figure to spheres. This makes it look more appealing.

Completed on Dec 9, 2017

We are currently in the process of making a color wheel. Our color wheel must be ridiculously small to have it working properly. A lot of ink and paper will have to be used to make the wheel work accurately. At first, we made a full circle, but then come to realize that the servo only rotates 180 degrees. If we wanted to have it so it would spin a full 360 degrees, then we would've had to take the servo apart and mechanically fix it, so it would allow us to. However, we chose not to do it like this because it would require not only new equipment, but also have the screws all over the place. We didn't want to break this servo because it is not ours. We wanted to make sure everything was taken care of and have it returned brand news. Back to the color wheel. We now must make a semi-circle and somehow attach it to the Arduino. If we get this then all we need to do is have Java or Arduino run the code. If we do it in Java, we must find a way to use Java and have it rotate the servo with Arduino code. Or,  we can use Arduino and find a way to call a variable created in Java and run it from there. The variable created in Java is called maxColorIndex. This variable allows us to get the text file from MATLAB with all the proportions and then have it sent to Java. Java then takes the proportions and sorts it from max to min all in an Array. Finally we can use this variable, and have it sent to Arduino ready to use.