

Dissertation - Electronics

Myo Armband with Interactive Model



May 2019

James martin sass

16005859

Contents

Contents Page 1

Plagiarism Statement 1

Chapter 1 - Introduction 2

Chapter 2 – literature review 3

Chapter 3 - Methodology 8

Chapter 4 - Results 17

Chapter 5 - Conclusion 18

Chapter 6 - References 19

Chapter 7 - Appendices 21

[Figure 1 - Myo Armband 3](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699106)

[Figure 2 - HTML Code 8](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699107)

[Figure 3 - CSS Code 9](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699108)

[Figure 4 - Global Variables 10](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699109)

[Figure 5 - Myo Poses Code 10](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699110)

[Figure 6 - GUI Code 11](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699111)

[Figure 7 - GUI Functions 11](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699112)

[Figure 8 - Sun Code 12](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699113)

[Figure 9 - Planet Ring Code 12](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699114)

[Figure 10 - Orbital Path Code 13](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699115)

[Figure 11 - Star Background Code 13](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699116)

[Figure 12 - Orbital Controls Library and Planet Axis Code 13](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699117)

[Figure 13 - Planet Orbit Code 14](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699118)

[Figure 14 - Stop Function Code 14](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699119)

[Figure 15 - Model Zoomed Out 15](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699120)

[Figure 16 - Model Zoomed In 15](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699121)

[Figure 17 - Plaent Selected 16](file:///C:\Users\user\Documents\UNI%203rd%20Year\electronic%20dissertation\Dissertation%20final.docx#_Toc8699122)

Plagiarism Statement

This project was written by me and in my own words except of the sections which are in quotations from published sources which are indicated. Al sources are listed in the references chapter.

Chapter 1 - Introduction

The main purpose of this dissertation is to explore gesture control in regards to technologies such as computers, televisions etc. assessing its potential applications for future technology.

Gesture controlled technology has many applications in multiple fields, such as medical where it could be used to recognise life threatening conditions or be used to assist the disabled. It also has a lot of potential in crating alternative computer interfaces where it could improve usability and even creativity.

The aim is to create an interactive simulation of the solar system, using the Myo band to make it capable of being interacted through gesture control. It will be capable of rotated around and zooming in and out entirely through hand gestures. The simulation will be large enough that the interactions can be properly explored to test their usability/ reliability.

This dissertation will discuss how the simulation was created and the gestures were implemented. It will discuss the testing, conclude the results and how the overall project could have been improved upon for a future test. There will be images to make the explanation easier to follow.

chapter 2 – literature review

This chapter of the dissertation identifies research and studies that have been conducted previously in regards to both the Myo Armband and Gesture Based Computing along with their history.



Figure - Myo Armband

The Myo Armband contains 8 Electromyography (EMG) sensors which are used to detect arm muscle movements of the forearm and hand gestures. It also includes a accelerometer, magnetometer and gyroscope which track arm movements.

Previous research

This section of the chapter reviews and summarises previous research performed by other people that is similar to this project.

1. “Taking a signal: A review of gesture-based computing research in education” (Feng-Ru Sheu, Nian-Shing Chen, 2014)

This research report looks into the uses of gesture-based computing technology in the education industry. They retrieved the data from over 3,018 articles created between 2001 to 2013 from 5 academic databases. 59 of the articles were analysed, identifying factors such as research methods, study disciplines, learning content, technology etc. Some of the finding were that the most commonly used research method was experimental design at 72.9%, which was followed by designed based research at 20.3%. the Nintendo Wii is the most commonly used gesture control-based device used in the experiments at about 40% and that it is most commonly used in special education. The most common learning topic was found to involve motor skills in some fashion. Some of the supporting gesture-based learning aimed to enhance learning effectiveness, especially for special education, which was found to be a common domain it was found to show up in according to the chosen reports. They that this study aims to provide opportunity to researchers from multiple to think on the future of gesture-based learning as it they believe it will become one of the primary trends in education.

1. “Using a gesture interactive game-based learning approach to improve preschool children's learning performance and motor skills” (Hsien-Sheng Hsiao, Jyun-Chen Chen, 2016)

Like the previous review, this one involves a research paper that looks into gesture control within education. The goal of this research project involves combining gesture-based technology with game-based learning into one to create something suitable for young children to use to help them develop motor skills, mainly focusing on agility and coordination. They call this approach ‘Gesture Interactive Game-Based Learning (GIGL)’. The ASUS Xtion Pro is a motion detecting device that also comes with software and developers’ tools to create gesture-based application. It was used in this project to create a virtual interactive learning environment. 105 participants were used for to test in a ‘quasi-experiment’. The age of the participants averaged out to 5.5 years old. The results of the project showed the participants of the GIGL experiment showed both a better learning performance and motor skills than those who only followed the game-based learning approach. The statistics show that there is a significant learning deviation between the two. The results show that the GIGL approach is a more effective learning method.

1. “Deep gesture interaction for augmented anatomy learning” (Ahmad Karambakhsh, Aouaidjia Kamel, Bin Sheng, Ping Li, Po Yang, David Dagan Feng, 2018)

Like the previous two research projects, this one focuses on how gesture control can be used in student learning, the difference being that this one is to assist in medical student learning, and is designed to work with augmented reality. Augmented reality is used in medical training due to the issue of having bodily organs in a classroom. Through the use of trained neural network (Convolutional) and an RGB-D camera, hand positions are both recognised and tracked in three-dimensional space. 11 gestures have been trained into the neural network and set as the standard inputs. They were tested in different ways and the results show there was a 97% accuracy in recognition. The design also allows for the addition of new gesture as necessary. There are two of the weak points that need to be improved upon in the future. The results show that the approach is more accurate than previous gesture recognition. It also shows the possibility of augmented reality being combined with neural networking as a field and its application in education. The first is the distance between the camera and the operator. The second is shaking the hand/ rapid movement.

1. “Light invariant real-time robust hand gesture recognition” (Ankit Chaudhary, J.L. Raheja, 2018)

Like the previous reports, this project works through gesture recognition. The difference is that this project focuses on improving gesture recognition instead of how it can be used in different situations. The project is based on increasing the accuracy of gesture recognition in differing light intensities without the need to wear a specialised glove. This project was going to use camera-based recognition and the use of an artificial neural network. They created a database of hand gestures and then tested them in differing lighting conditions, differing greatly to get proper results. An ANN as trained using images from different sources such as online searches and manually collected. 14 different images were used for each gesture during training of the ANN, 6 gestures were used for this project. The tests were also performed on multiple people to get a wider testing base. The results were that the gestures were correctly identified with a 92.86% accuracy.

1. “Low cost remote gaze gesture recognition in real time” (David Rozado, Francisco B. Rodriguez, Pablo Varona, 2012)

Like the previous reports, this project works through gesture recognition. The difference is that this project focuses on eye gesture recognition rather than had based ones. Using a pattern recognition algorithm called Hierarchical Temporal Memory (HTM) for this project, it was adapted to work with real time changes of eye gestures. This change allows for reliable discrimination of intentional/ non-intentional gazes. The results show a 98% accuracy of recognition for 10 categories of gaze gestures. There was a low rate of false positives and the determination of the gazes was completed in an acceptable amount of time. The results support the use of gaze gesture-based hardware as a field for thing like smartphones/ TVs, projected displays and desktop computers.

1. “Motion-towards-each-other-based hand gesture initialization” (Zhiquan Feng, Bo Yang, Na Lv, Tao Xu, Jianqin Yin, Xiuyang Zhao, Shichang Feng, 2015)

The aim of this project is to reduce the user’s operational burden and improve the computers response to the users input to at least acceptable levels in the creation of a 3D hand model from a 2D image and have the algorithm recognise hand gestures and replicate them. The system works by camera, it recognises intention to start through the use of its hand gesture sensing system algorithm. From there it works with the user until a pre-defined meeting point is achieved. This proposed approach effectively lowers the burden on the user, both cognitive and operational. This system was effectively applied to several other systems. During a test 4 of the gestures was each tested 1,500 times, two of the were recognised 100% of the time, the other two gesture were recognised 99.8% times.

1. “Recognizing expressions from face and body gesture by temporal normalized motion and appearance features” (Shizhi Chen, YingLi Tian, Qingshan Liu, Dimitris N. Metaxas, 2012)

This project works with gesture control of both the face and the body of the user. It works through feature extraction of the face and the body, feature representation and affect recognition. This work is only focusing on the upper body. By combining multiple MHI-HOG (Motion History Image – Histogram of Oriented Gradients) and Image-HOG together. MHI-HOG captures motion direction from a pre-defined point of interest while Image-HOG captures appearance information from the corresponding point. Both of these work with temporal normalisation which solves the issue with time resolution from the affect recognition in the video. The experimental results from this project show that the performance is comparable with state-of-the-art works. There are significant improvements in recognition accuracy.

1. “Design and implementation of an eye gesture perception system based on electrooculography” (Zhao Lv, Chao Zhang, Bangyan Zhou, Xiangping Gao, Xiaopei Wu, 2018)

This project is similar to project ‘5)’ in that it is gaze gesture based. The aim of this project is to help people with motor diseases communicate if they are suffering from verbal/ non-verbal communication degradation. The way this project is going to help is that they will create an eye gesture recognition system that works through electrooculography (EOG), a technique that detects signals called electrooculograms, which can be used as a way to record eye movements. This is combined with a feature extraction algorithm, which is set to detect eight specific movements of the eye. The experimental results of this project show that there is over 95% recognition rate from over 10 different subjects, which shows that the system has a good performance rate. The research believe that this system can be combined with a video-based system to improve its performance.

1. “Human hand gesture based system for mouse cursor control” (Horatiu-Stefan Grif, Trian Turc, 2018)

Like the previous report, this project is designed to help those with disabilities, in this case people who are wheelchair bound. Unlike the previous project, this one is not gaze gesture based but hand gesture, which makes it more connected with this dissertations project. This project focuses on using hand gestures to move a mouse cursor. A camera is placed over a blue mat, the hand is placed on the mat and is detected from there by performing the correct gestures. The reason for the blue mat is to provide contrast to the hand, making it easier for the gestures to be seen. The gestures chosen were picked because the hand postures are natural and relaxed ones, making it better/ more comfortable for the user. The system was tested at different levels of luminosity and proved that it was still capable of detecting the hand gestures in all but one scenario.

1. “User-adaptive hand gesture recognition system with interactive training” (Attila Licsár, Tamás Szirányi, 2005)

This project is another augmented reality-based project. This project is designed to recognise hand gestures, teach new user the hand gestures and learn them new users and how they perform the gestures so it can improve its gesture recognition. The augmented reality side of the project come in with projections that can be interacted with and manipulated by the hand gestures. For example, the report from the project uses the example of a projected presentation during a meeting and how they are normally manipulated by mouse/ keyboard; they say it would be more comfortable for the user to be directly interacting with a projected image. During the testing of the system, only one user trained all of the gestures and all following users went on to retrain gestures that they had low recognition rates with. The tests show that the system is easy to use, and that performance of recognition has increased, especially when the user has trained the systems gestures.

1. “Tactile feedback enhanced hand gesture interaction at large, high-resolution displays” (Stephanie Foehrenbach, Werner A. König, Jens Gerken, Harald Reiterer, 2009)

This report is about testing perceptions, in this case adding a physical perception from human to computer interactions due to it usually being limited to visual/ audial. This is done by from combining hand gestures with tactile feedback to simulate physical interaction. Unlike all of the other reports discussed within this dissertation, the results of this project aren’t an overall positive. The testing for this project was done in front of large wall sized display equipped with 6 cameras that together cover the front area of the display. The test included 20 participants who, when being tested, where placed in front of the display and had their hand equipped with the glove that provides the tactile feedback performed the necessary gestures for the test. The results showed that there weren’t any noticeable benefits for the tactile feedback, in fact, there was an increase in errors with the tactile feedback during part of the test than without. They recommend further testing to discover the usefulness of tactile feedback.

1. “Touch screen gestures for web browsing tasks” (Sabrina S. Billinghurst, Kim-Phuong L. Vu, 2015)

This project combines hand/ finger gestures with touchscreens. The test involved 20 people, some phycologists some engineers, who had to design hand gestures based of an action they saw performed on a screen. Afterwards, participants were asked to rate how well they feel the gesture the created matched the action and how easy it is to perform. These procedures were repeated until all actions were completed. The results show that most of the gestures created used the participant dominant hand, index finger and a signal motion. When the task involved a tablet device rather than a phone the gestures involved all five fingers rather than a single. There was no noticeable different between the phycologists and the engineers.

1. “Direct hand pose estimation for immersive gestural interaction” (Farid Abedan Kondori, Shahrouz Youseﬁ, Jean-Paul Kouma, Li Liu, Haibo Li, 2015)

The purpose of this project is to create a more immersive form of gesture interaction with computers. They describe the two main issues to tackle with this project as gesture recognition and gesture pose sestimation. These two issues when solved will create dynamic gesture recognition. This project allows the user to recognise hand gestures and manipulate the object on screen accordingly in real time. The image the object is on comes from a live camera feed, making it look like the object is in reality. The designed system works very well with only one issue, that being extreme finger movement. This is because the finger motions can affect the tracking algorithm for the system because they are not treated separately.

1. “Finger-stylus for non touch-enable systems” (Ankit Chaudhary, 2017)

The goal of this project is to create a way to draw on a monitor with hand gestures without actually touching the monitor as an attempt to replace the stylus with the more natural finger/ hand. The finger is the input device and acts like a paintbrush and a stylus; the user ‘draws’ in front of a camera. This was designed to work through a combination of fingertip extraction and motion tracking, created in MATLAB. The advantages of this is that there is no need to wear specialised gloves or the need for touch screens, reducing costs. Another advantage is that it is simple to use. The designers of this project say that in the future they would want to implement 3D to it for more advanced applications.

1. “Low cost remote gaze gesture recognition in real time”

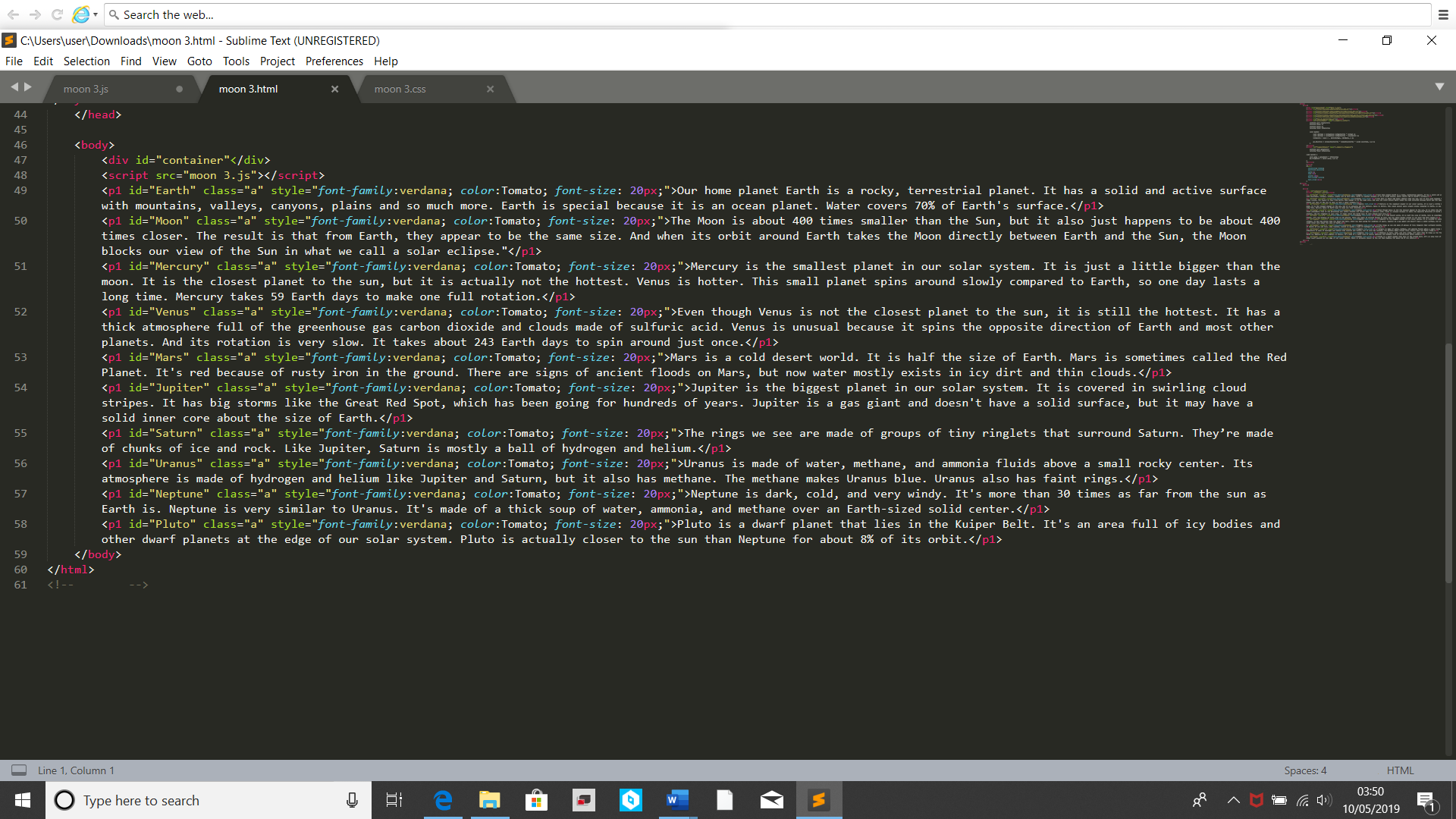
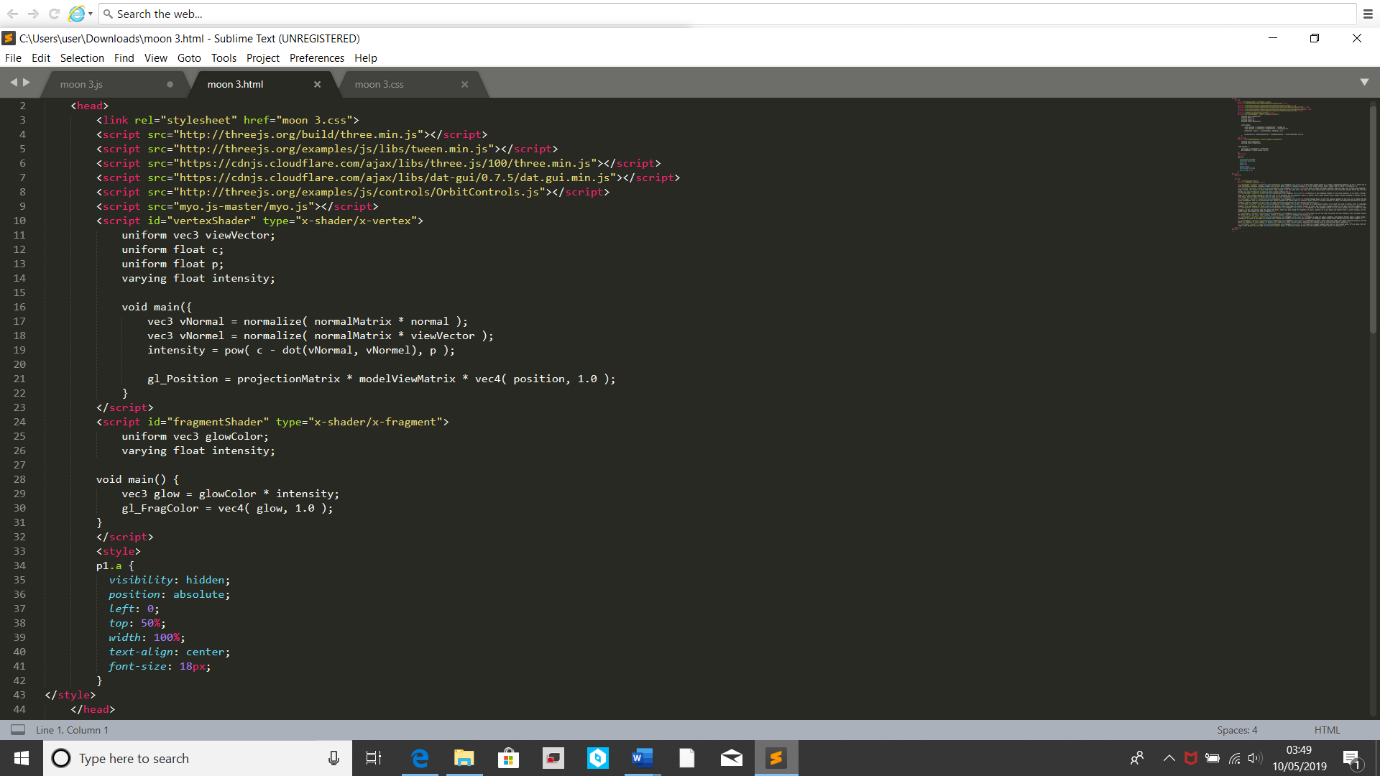
The purpose of this project is to combine the information gained from both gaze gestures and the perspective from a hand mounted camera (placed beneath the wrist) to create a system that can detect hand movements using only objects contained within a scene even without any markers on said object. Not only that, it is possible to use this data to predict which object is going to be grasped before the user actually grabs it. The results of this experiment show that the users grasping and the targeted object can be predicted from the information gained from the combination of gaze gestures and the hand camera.

Chapter 3 - Methodology

In this section of the dissertation, the methodology behind the project will be explained. How the coding of the project works with images of the code itself to go alongside it. The code of the project is split into three different sections that work together:

HTML:

Figure - HTML Code



HTML stands for HyperText Markup Language. It is a programming language used when creating webpages. It is used to describe the structure of the contents a HTML file. Above are two screen shots of the HTML section of code. At the top are a list of the different programming language libraries that are used in the creation of this project, including the JavaScript Myo library. Also included in this is the CSS file (discussed below). In the ‘Style’ section of the screenshot you can see how the text should appear in the project. It declares that when the project is started up it should be hidden and should only appear in the centre of the screen at a size of 18px when not. In the second screenshot is the ‘Body’ section of the HTML, this where the JavaScript part of the code is called in (discussed below). Below that is the part where all the texts of the project are written. These texts are the same ones that are effect by the ‘Style’ part. They only appear in when select by the GUI in the main project. There colour and font are listed before the actual text themselves.

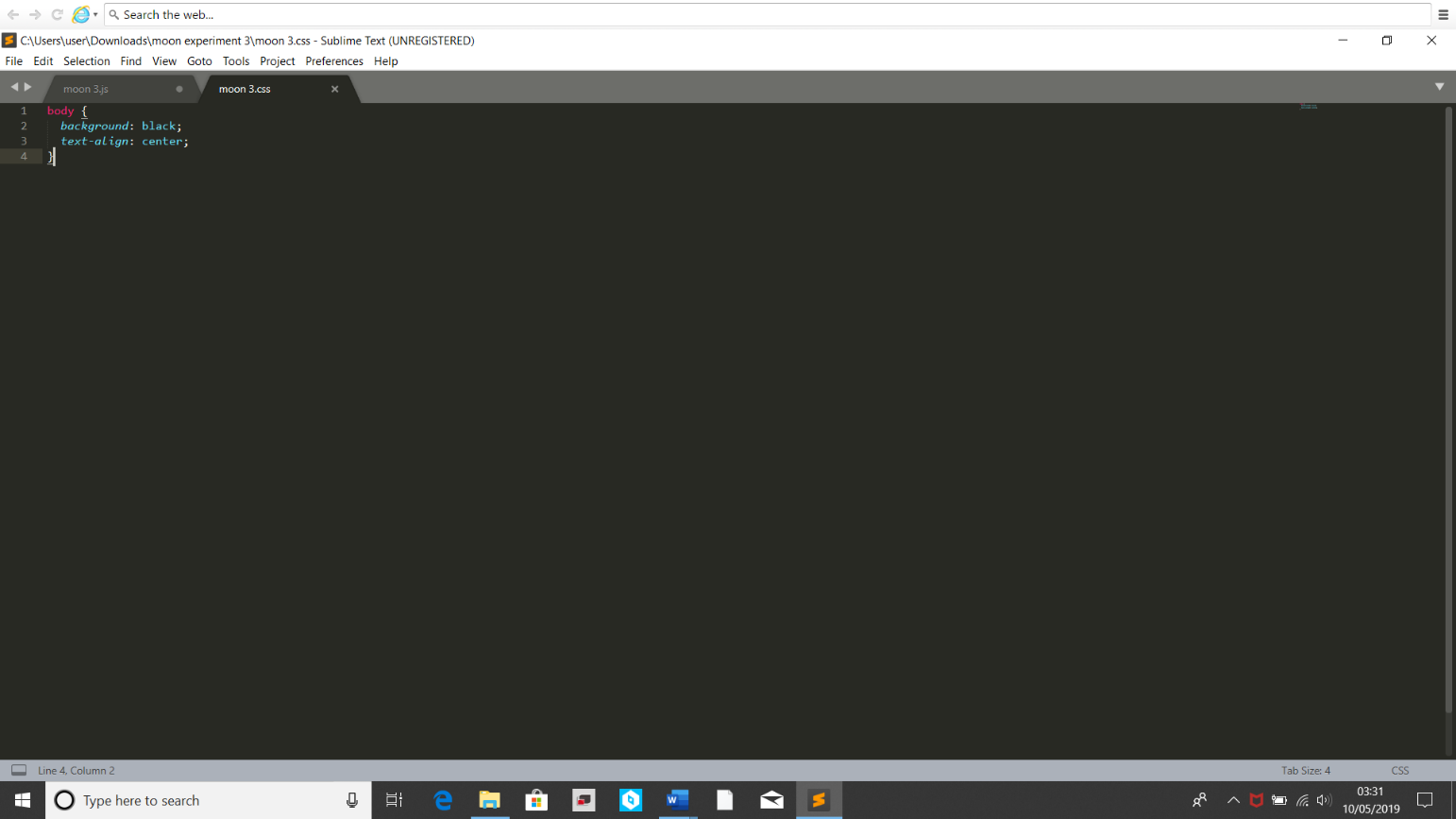
CSS:

Figure - CSS Code

CSS stands for Cascading Style Sheet. It is used in combination with HTML to describe the way the elements are to be laid out within the chosen media, in this case being the screen through Microsoft Edge. Above is a screenshot of the CSS section of this projects code. It shows that the project should have an all back ground and all text should be centred on screen. It is an incredibly simple code due to the fact that the real background used in this project is an image of stars, as such this black background is just redundancy in case the user zooms out of the star image, without it, when the user zoomed out too much they would suddenly see a white background, which would be jarring.

JavaScript:

JavaScript is a programming language that is normally used to create interactive web browsers. The reason this was chosen was because it not only was this a well-used and liked programming language for creating web pages and as such had a lot of libraries out there to support it, but also because one of those libraries was a library for interacting with the Myo band.

Below, like previously, there will be screenshots of the code and descriptions of what they do. Unlike previously though, due to the sheer size of the JavaScript code, the screenshots will only show parts of the code due to repetitiveness of it. The explanations will still cover it all.

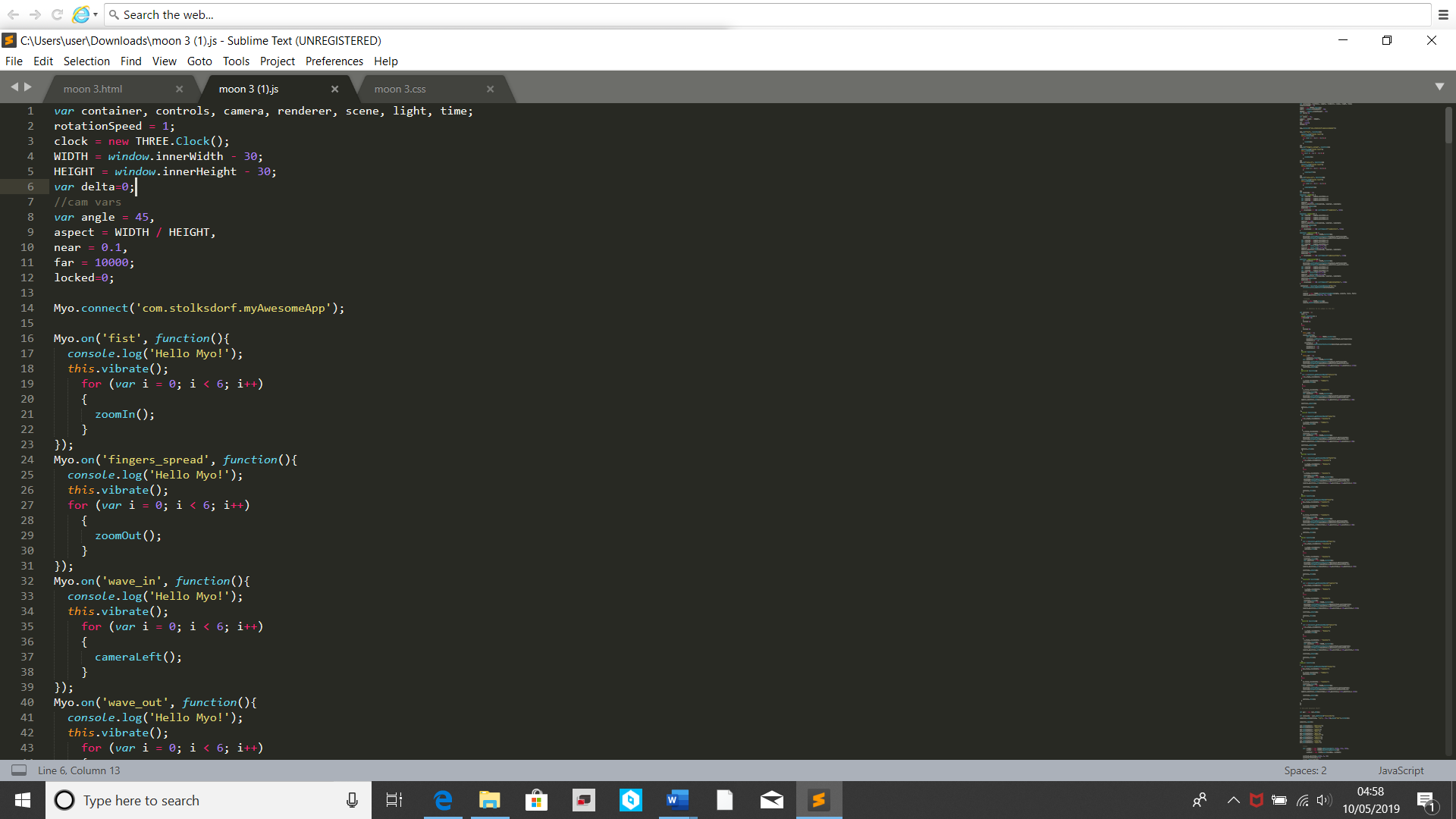


Figure - Global Variables

This image (Fig 4 )shows the global variables being established.

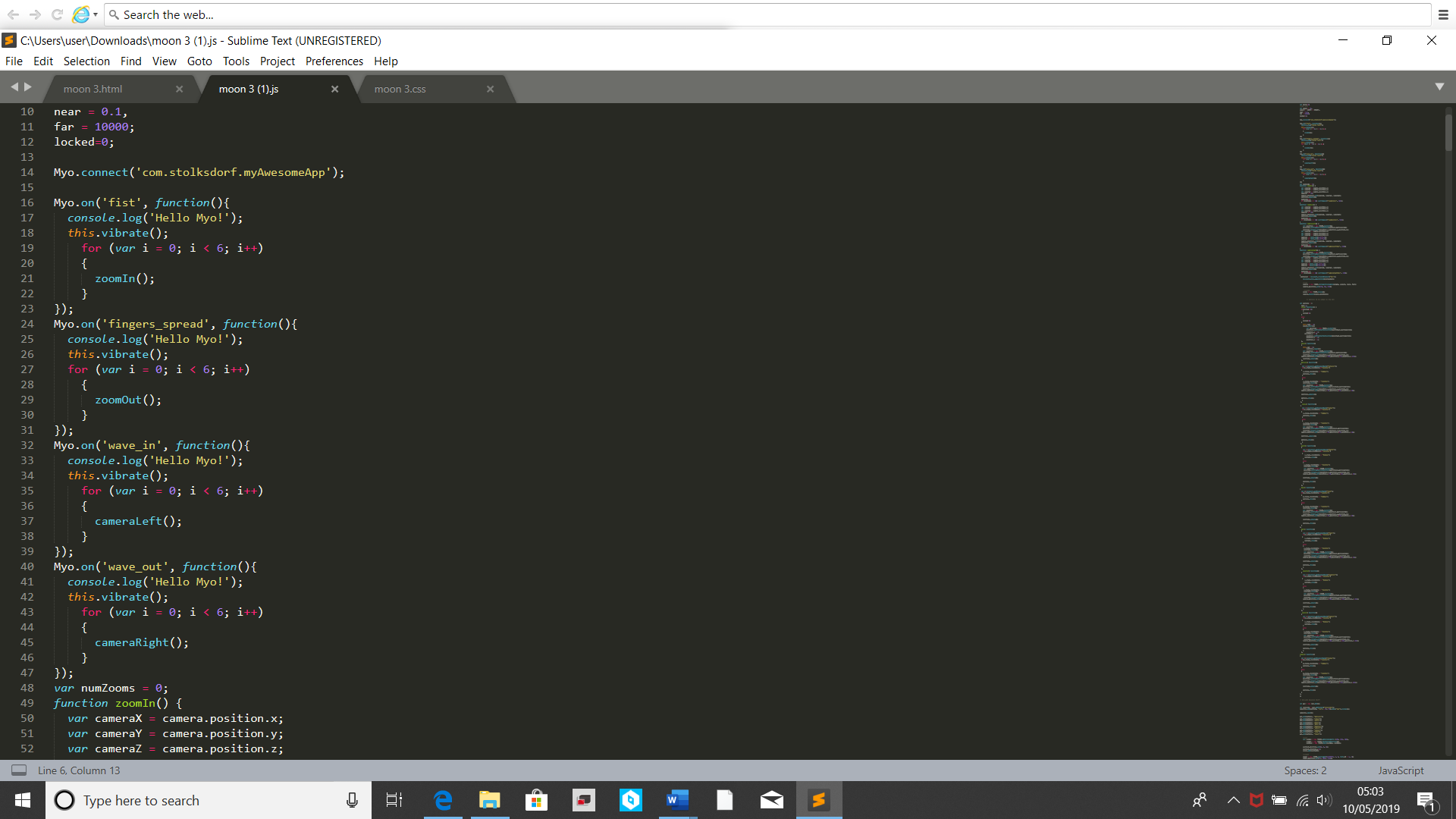


Figure - Myo Poses Code

This section of the code shows the Myo poses being established. After calling the Myo Library. Four poses were chosen, ‘Fist’, ’Fingers Spread’, ’Wave In’ and ’Wave Out’. Each of these poses were coded to have an effect on the solar system model. ‘Wave In’ and ’Wave Out’ are able to move the camera to the left and right, respectively, in a circular form. The poses ‘Fist’ and ’Finger Spread’ Zoom in and out the camera respectively. When a pose is detected, the Myo band will vibrate and the relevant function will be called to affect the cameras placement within the model. The camera will keep moving for as long as the pose is held.

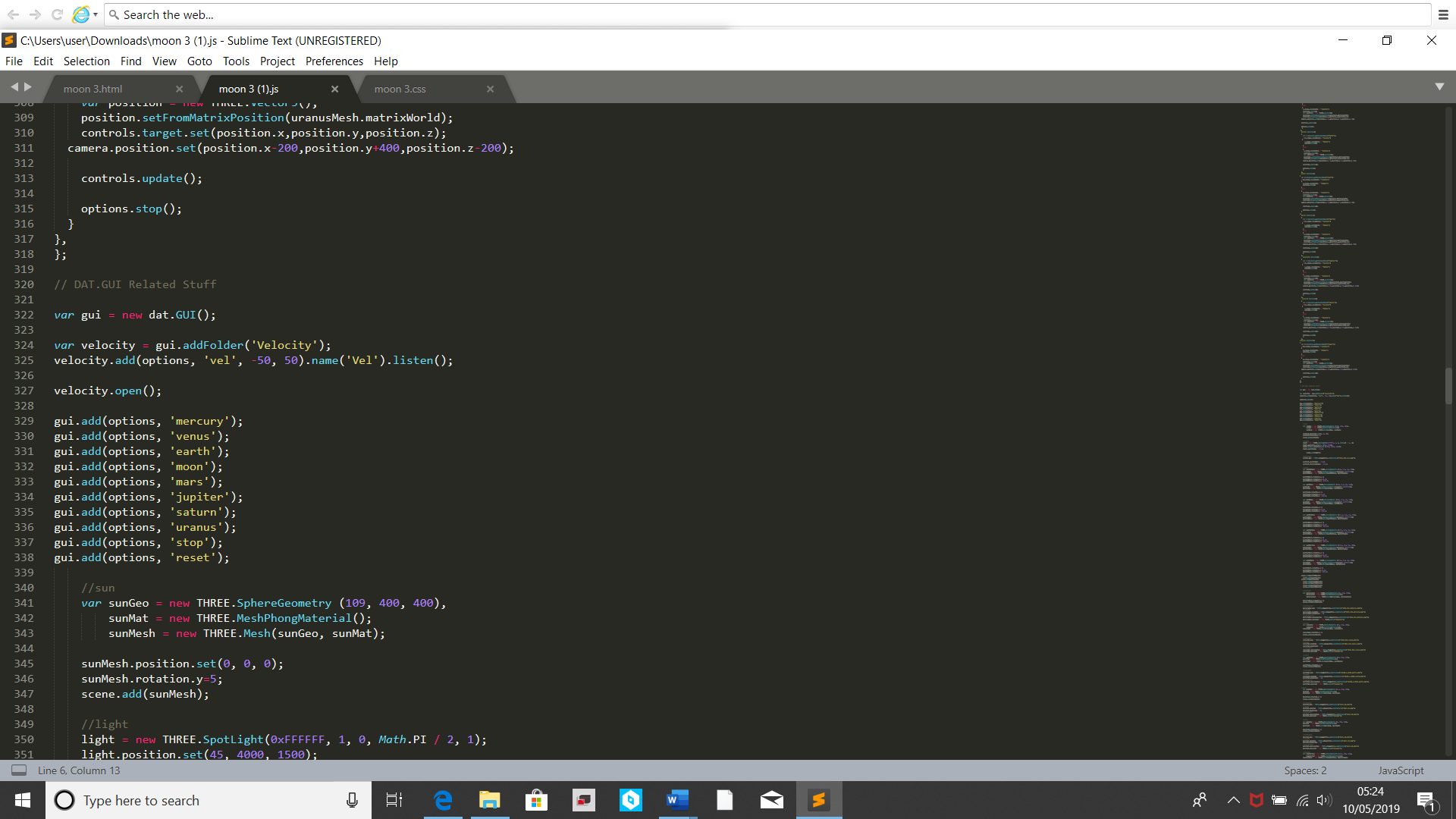


Figure - GUI Code

This section of the code makes use of the ‘Dat GUI’ library called in the HTML section of the code. This creates a GUI in the model that effects the model depending on the option selected. The code above shows a list of all options in the GUI. The ‘Velocity’ options is a bar that can be dragged to effect the speed the planets are moving at. It can even cause the planets to move in reverse.

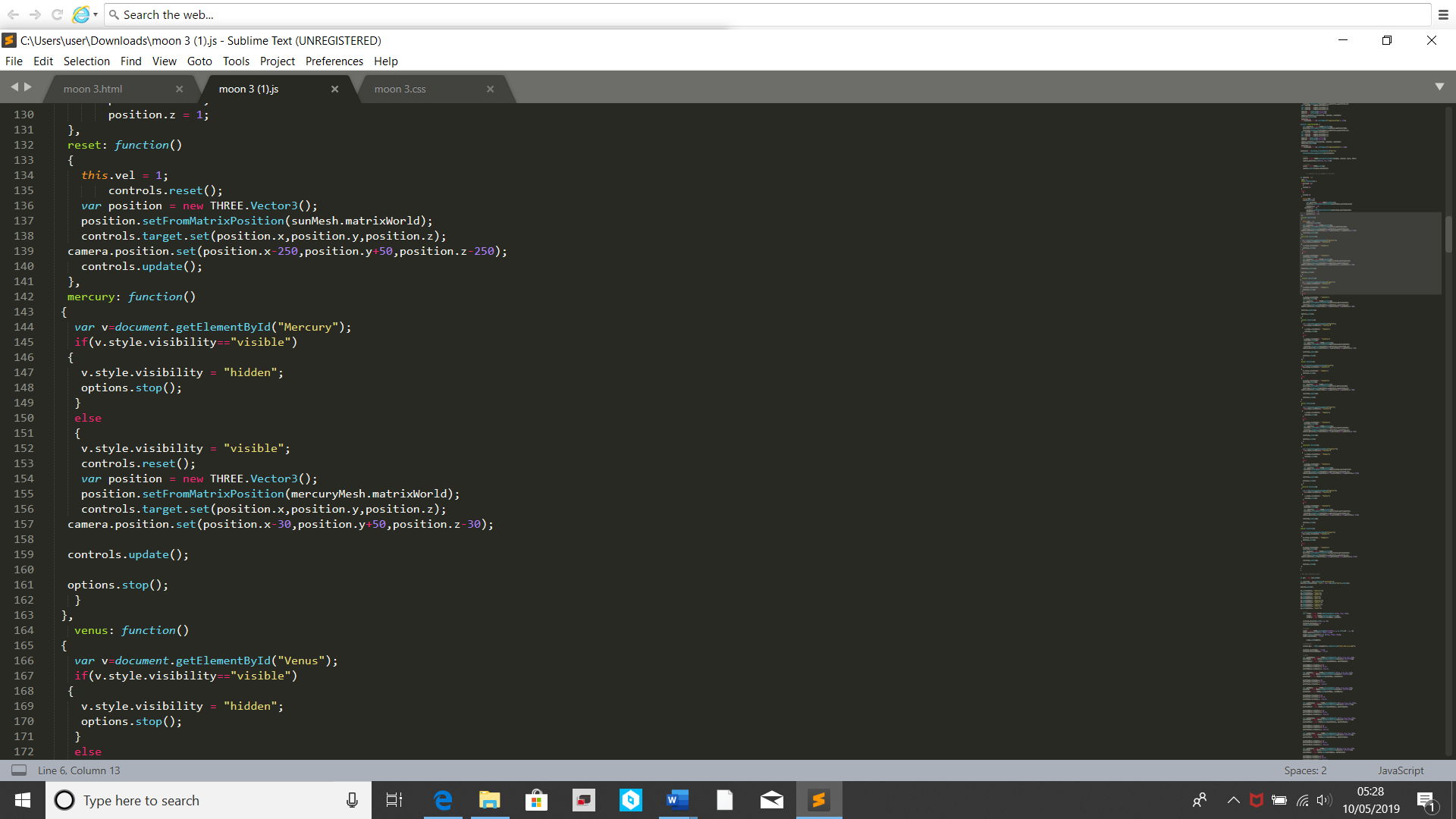


Figure - GUI Functions

This section of the code directly relates to the last. It shows both the ‘Reset’ and the ’Mercury’ GUI functions code. the ‘Reset’ works by resetting the camera position from where it is to where t originally starts when the model is opened up. The Mercury code works by calling the ‘Stop’ GUI function, which causes all object/ planets to stop moving. Once it’s done that, it finds Mercury’s current X-Y-Z position and moves the camera slightly away from it but pointing towards it. It then reveals the Mercury text created within the HTML code. The GUI code for the other planets is the same except for the planet they are affecting.

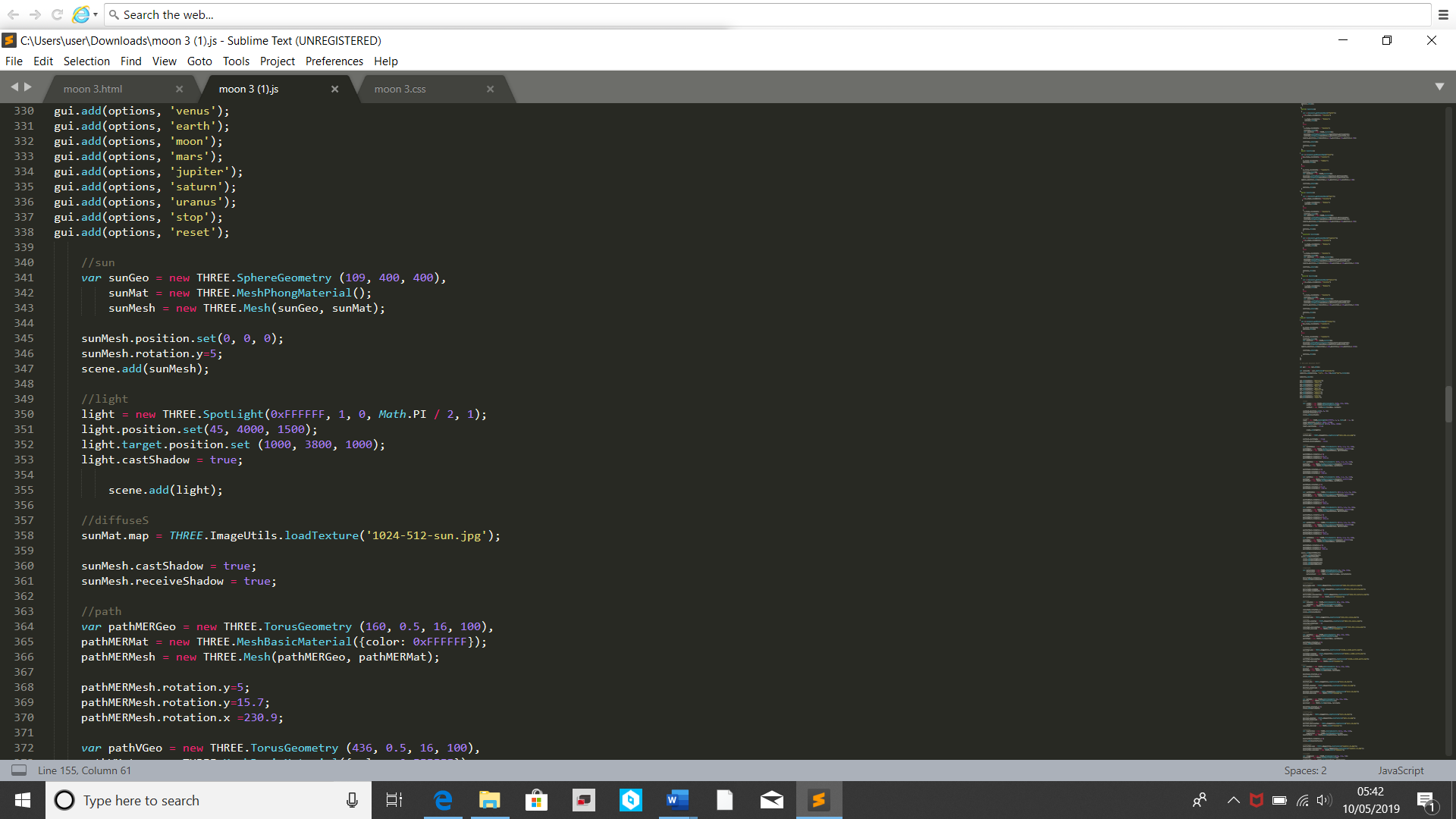


Figure - Sun Code

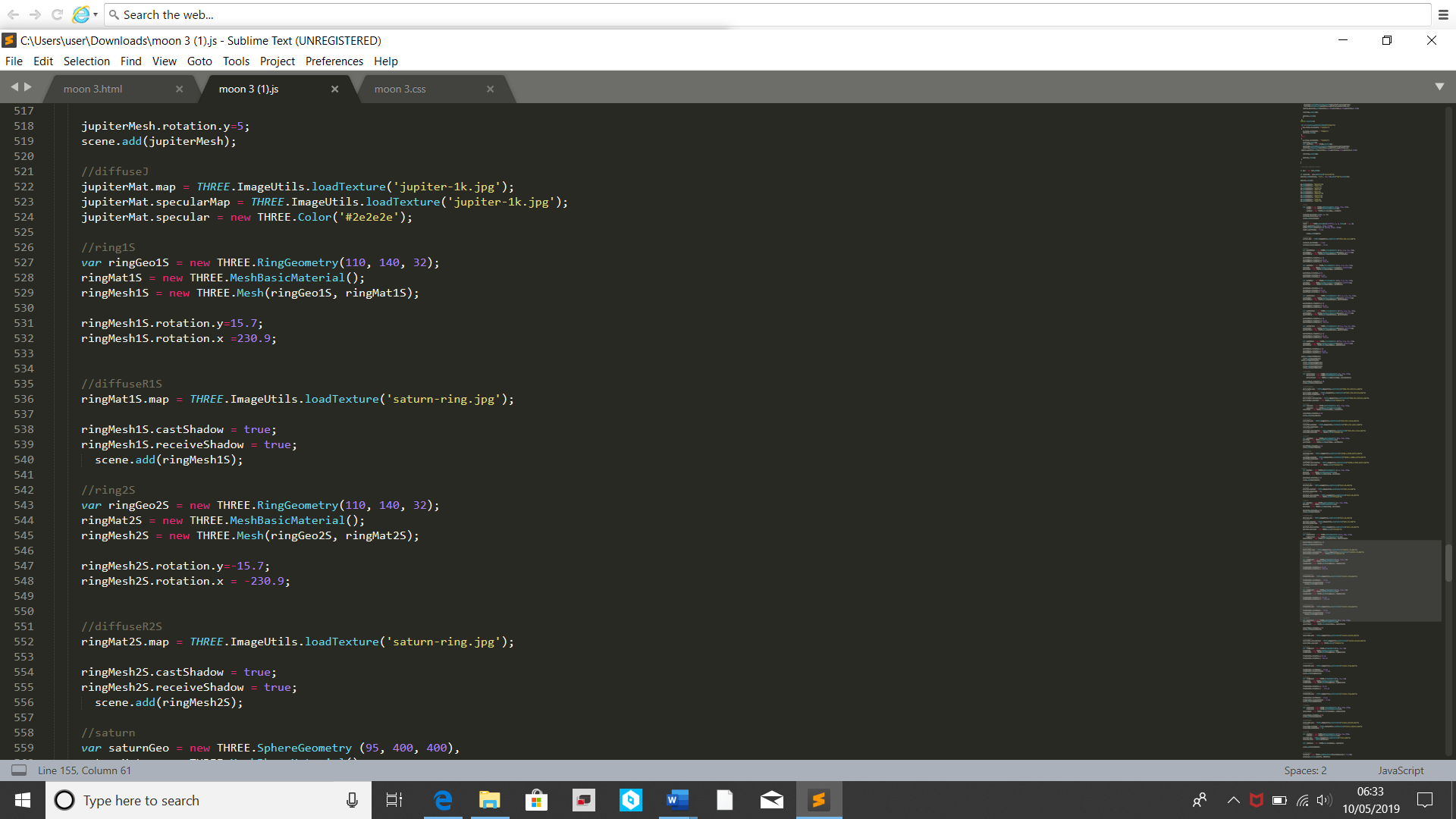
The above code shows the creation of the Sun object in the model and the light source that lights the model up. It creates a sphere object of a determined size, sets it position at the centre of the model and wraps an image around it from the same file that the JavaScript code is in. the image is that of the sun bursting with solar flares. The light source is similar to a torch shining down on the Sun and spreads out from there. The code allows for the light source to cause shadows. All the other planet codes are similar to the Sun code, the differences being that they are placed in locations further out from the centre and have different images wrapped around the based on the planets they represent.

Figure - Planet Ring Code

The above code shows the creation of the rings of Saturn in the model. It works similar to the Sun object but requires a different shape, Ring, and requires two objects be created and placed back to back with each other. They are sized so that Saturn can fit between them and then they are placed around Saturn. The rings of Uranus are created in the same way.

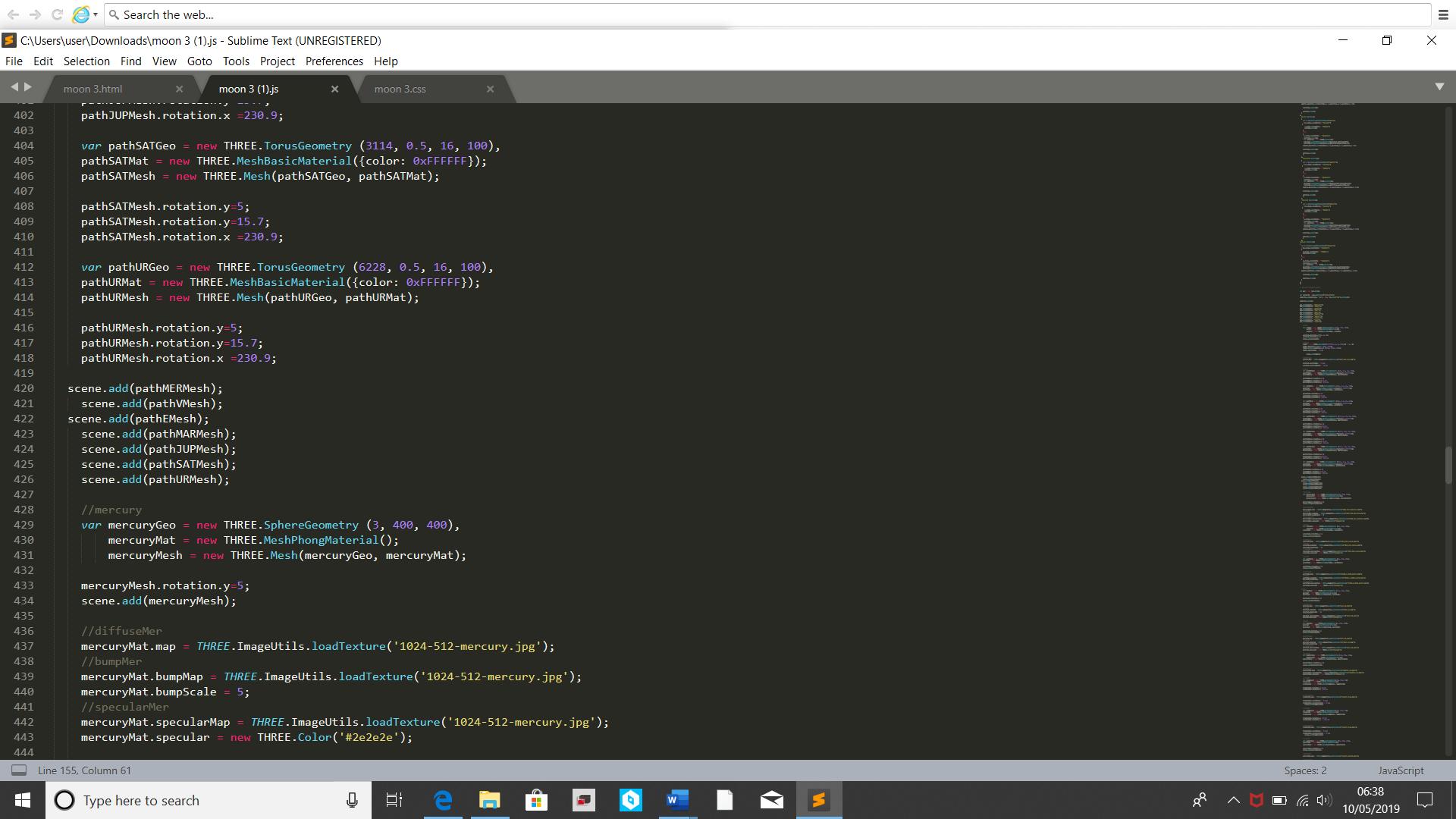


Figure - Orbital Path Code

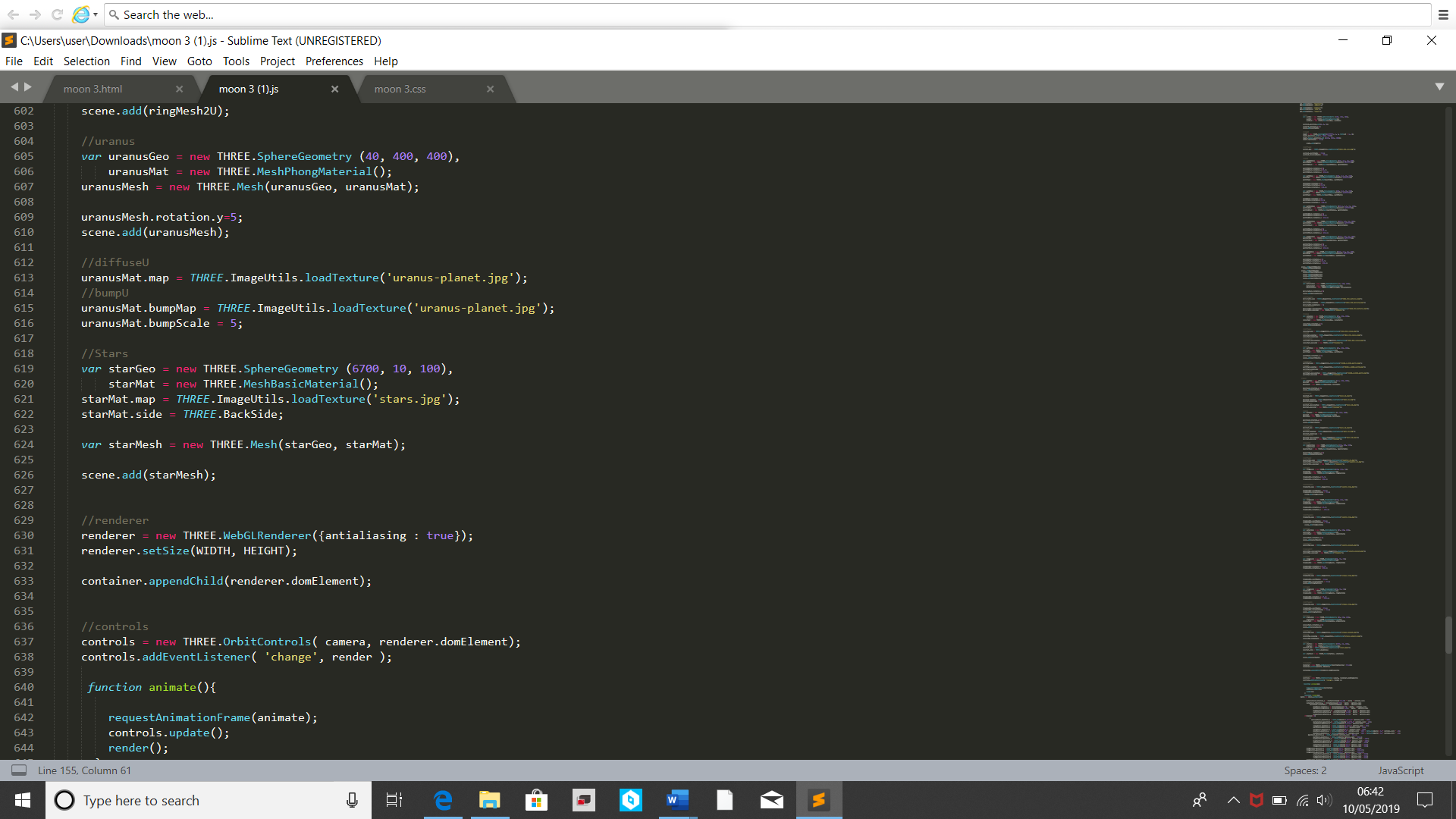
The above code shows the creation of object variables called paths. Each one is a glowing circle, created using Torus geometry, that outlines the orbital path of each planet. At the bottom you can see each path being added into the scene.

Figure - Star Background Code

The above code creates the starry background for the model. It works by creating a large sphere object similar to the planets. Then it places an image with the interior of the object instead of the outside. All of the rest of the model happens within this sphere, creating the illusion that it is happening inside space.

All objects and camera work created in this model where built using the library ‘Three.js’.

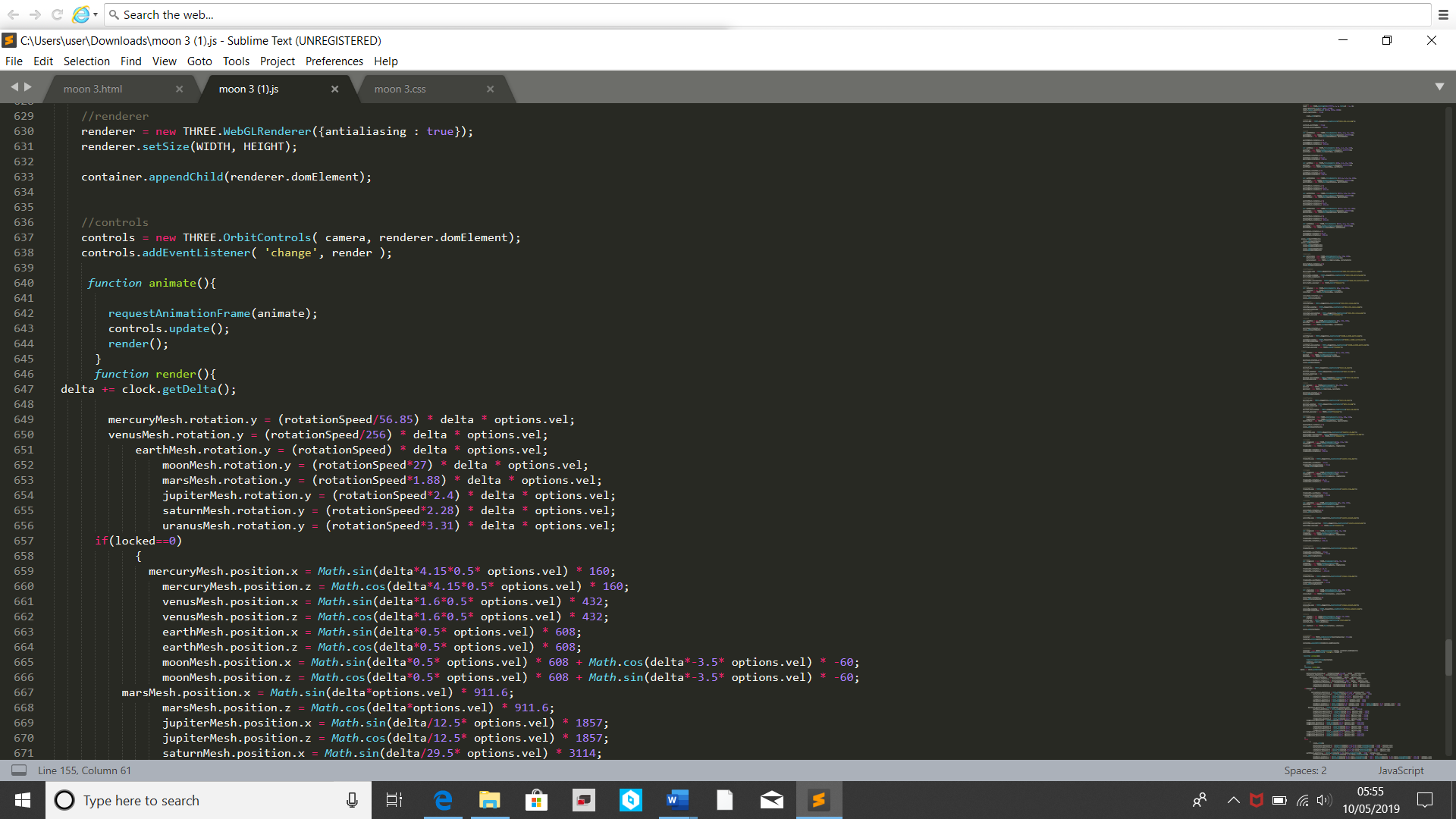


Figure - Orbital Controls Library and Planet Axis Code

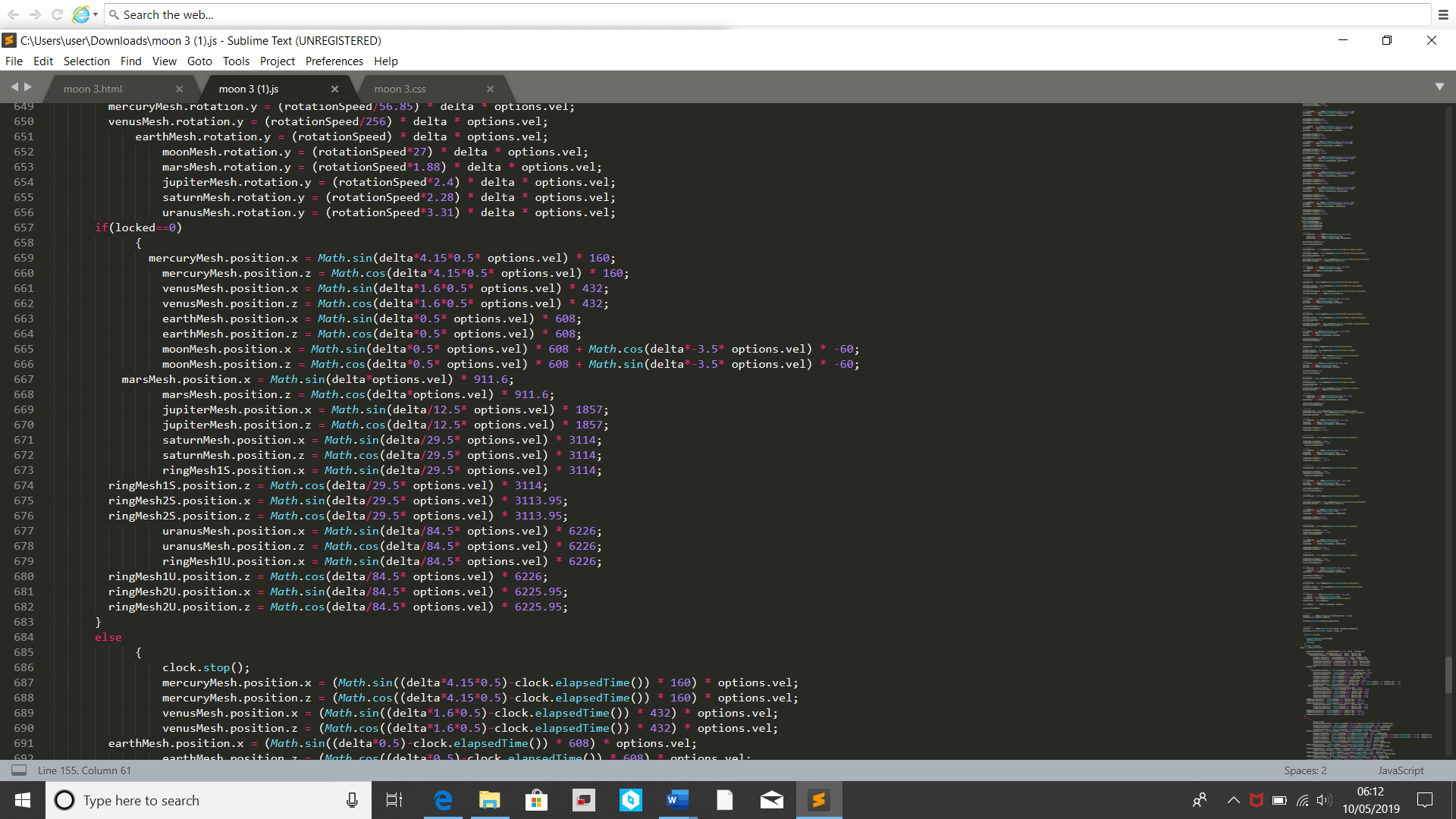
The above code shows three different sections. the first is the controls. This makes use of the ‘OrbitControls’ library, which allows for much easier interaction for the camera. The second it the ‘animate’ function that gets called at the end of the JavaScript code, which allows for the model to keep moving. The third is the code that causes the planets to spin on their own axis. It is done by using the ‘delta’ variable to continuously alter their Y position. Each one has its orbit altered (purple code) to match the real orbit of the planet as close as possible. The ‘options.vel’ is part of the velocity GUI that effect the overall model speed.

Figure - Planet Orbit Code

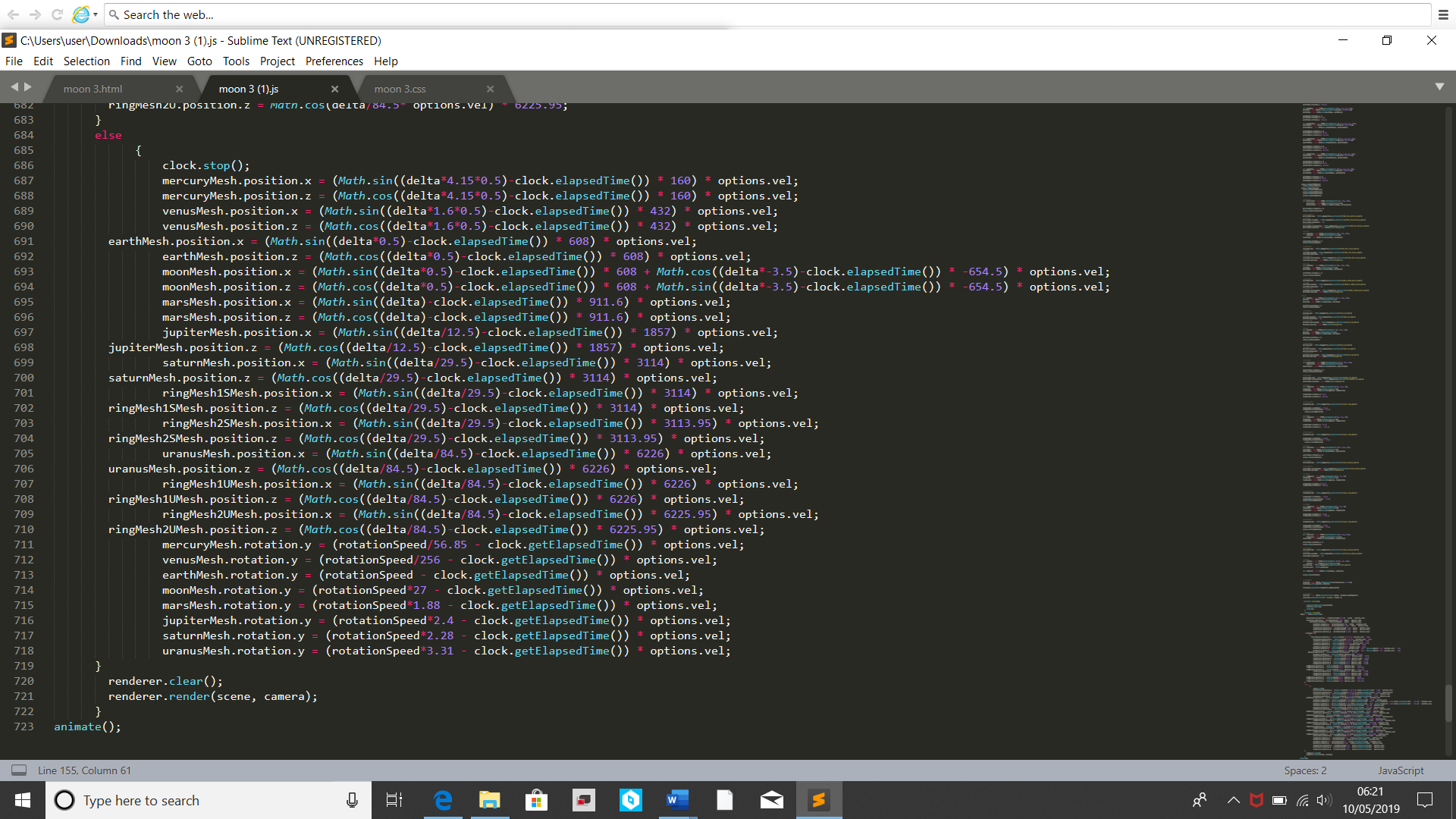
The above code is where the planets orbits around the Sun are created. It is done by using the ‘delta’ variable to continuously alter both the X and Z positions of the planet objects. The purple code parts are where each planet is individualised. The purple code alters the speed of the rotation around the Sun to make it closer match the real-life speed of these planets. Like previously, the ‘options.vel’ is part of the velocity GUI that effect the overall model speed. The ‘if locked=0’ is for the ‘Stop’ GUI function to come into effect in the next bit of code. the reason the Moon has longer lines of code compared to the others is to allow it to rotate around the Earth while rotating around the Sun.

Figure - Stop Function Code

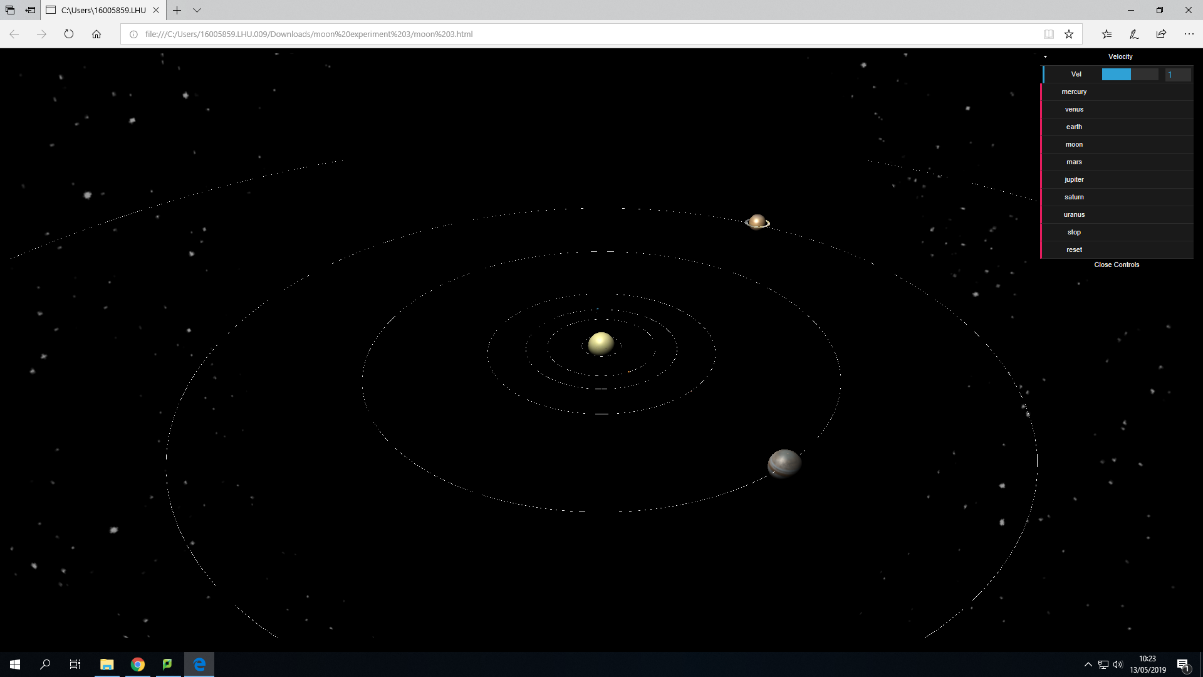
This section of the code only comes into effect then the ‘Stop’ GUI function is called. First it stops the JavaScript clock from continuing to count, which stops the delta variable, which stops the model objects from moving. When the ‘Stop’ function is over, the ‘clock.elapsedtime’ minuses the time that has passes between the clock stopping and starting again. This allows the planets to continue moving from where they left off. Without the elapsed time from being removed, then the planets would have jumped across the screen, like they teleported. At the very bottom of the code, the animate function is called to start the model moving when opened.

Figure - Model Zoomed Out

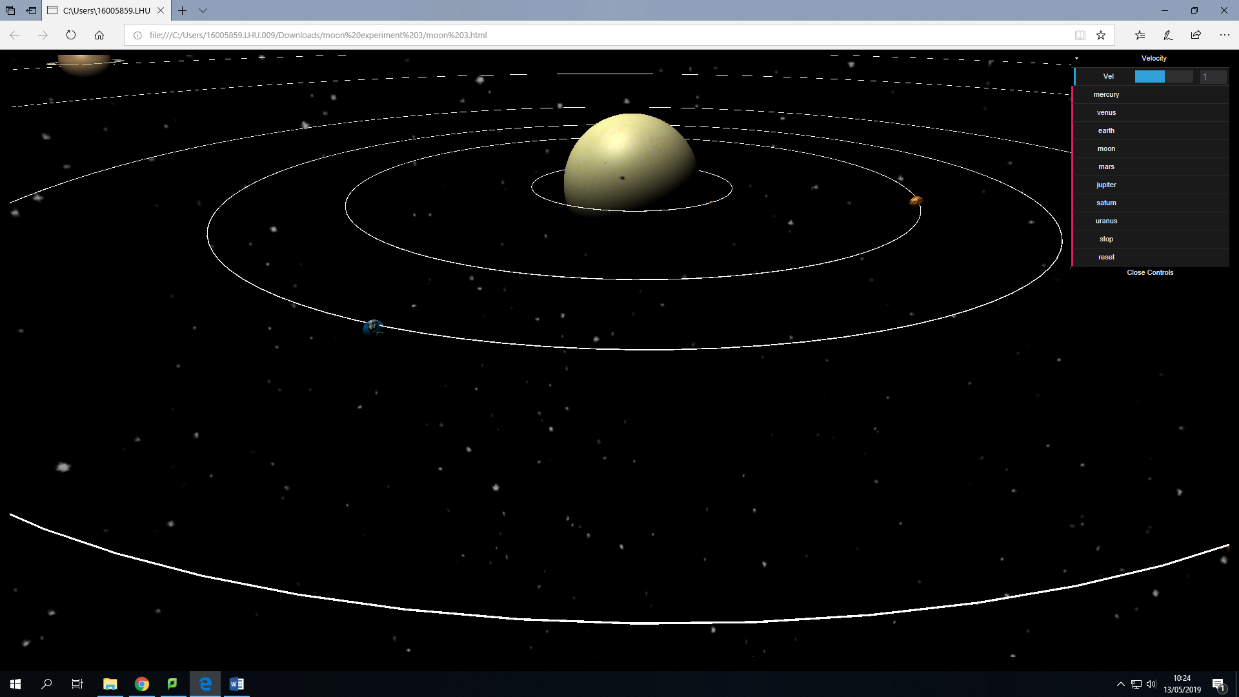
This image shows the model of the solar system itself. In this image you can see most of the model as it has been zoomed out. You can see the larger objects at this distance due to their size but cannot see the smaller ones. On the top right you can see the GUI which allows for the capability to alter the velocity of the model and the ability to zoom into a specific planet and see a basic summery. You can also see the ability to stop or reset the model.

Figure - Model Zoomed In

In this image you can see the smaller planets a lot easier due to it being zoomed in. it is also easier to see the orbital rings which show where the planets orbit around the sun.

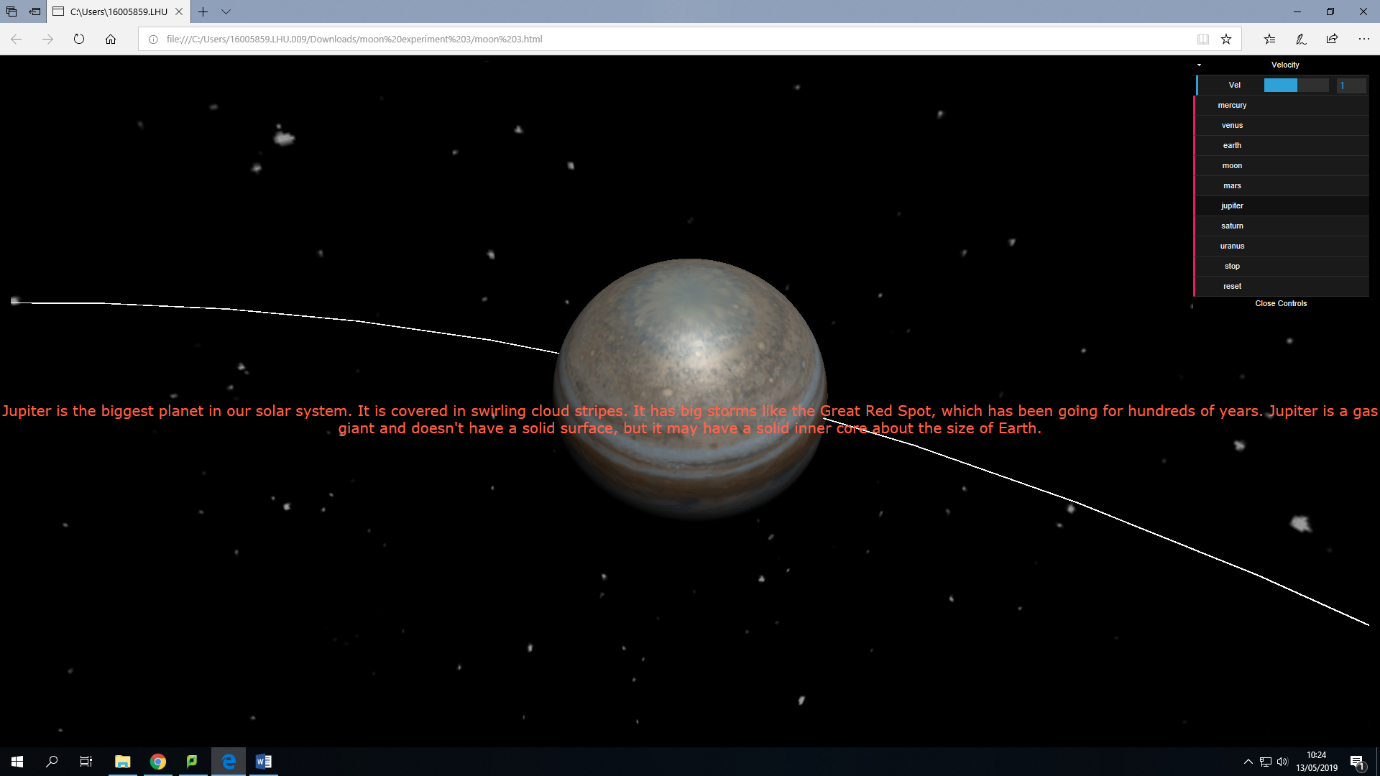
This image shows what it can look like when you select one of the planet objects from the GUI. The model freezes and the moves the camera to a point where you can see the selected planet up close. Then it displays basic information about that planet in a colour that makes it easier to read on the foreground.

Figure - Plaent Selected

Chapter 4 – results

While building this some issues occurred such as being unable to spread the image across both sides of the planet rings for both Saturn and Uranus, this was solved through creating two and pressing them back to back.

Upon on testing the finished model, most of it worked very well. The planets all orbited on their own axis and around the Sun. they all orbited in the correct directions. The orbital paths all appeared in the correct positions. All of the GUI options appeared and all worked as planned. The Myo control all worked the way they were supposed to.

The issues with the Myo side of the project as that it either detected the wrong pose or relatedly detected the same pose multiple time instead of once. This caused the camera to move in erratic ways. Since this issue is with the Myo band, there is little that could be done to fix this. Another issue is that it doesn’t load in Google Chrome, no matter what and upon loading up in Microsoft Edge, it can crash easily. It works much better in Microsoft Edge after lowering some sections of the code and removing unnecessary parts but it still crashes easily. I believe this is due to a combination of so many moving objects and Microsoft Edge itself, but since it doesn’t work with Chrome, it is the only option.

Chapter 5 - Conclusion

This research is aimed to create a simulation of the solar system in which to test/ explore hand-based gesture control within to assess the potential applications of the technology. This is done by allowing the manipulation of the created simulation through the gestures, which are detected using the Myo armband. The simulation was created in JavaScript code and included planets from Mercury to Neptune, including a Sun for them to orbit around. The simulation was quite realistic by having the planets orbit at accurate speeds, relative to the Earth model. Once the simulation was set up, the Myo code was added in, allowing gestures to affect the simulation. Four gestures were used to affect the simulation, not including the ‘Double-Tap’ gesture the Myo band recognises to start detecting hand poses. For zooming in and out, the ‘Fist’ and ‘Fingers-Spread’ poses where used, respectively, while moving left and right were controlled by the ‘Wave-In’ and ‘Wave-Out’ poses. These multiple poses, combined with the size of the simulation, allowed for great testing of the Myo Bands ability to detect hand-gestures, and how well they interact with the computer and simulation itself. The hand gesture inputs were detected by the Myo band and did affect the simulation as planned. Each off the gestures did affect the simulation in the correct way, rotating the camera in the right direction and zooming in and out.

There was one downside, this being that the Myo band sometimes incorrectly identified a gesture as being a different one, or that a gesture was being made multiple times when it was only being made once. This caused the camera in the simulation to move either erratically or in the wrong way occasionally. Another issue though not a big one was that the camera movement was stuttered slight when being moved based on the poses detect. This issue is due to the model itself and could only be improved based on better hardware for the computer.

If this research where to continue then the best way forward would be to find out what causes the Myo armband to detect incorrect or multiple poses and correct it, this is important as inputs based on detected poses are easily affected by this problem.

Chapter 6 - References

Ahmad Karambakhsh, Aouaidjia Kamel, Bin Sheng, Ping Li, Po Yang, David Dagan Feng. (2018). *Deep gesture interaction for augmented anatomy learning.* Available: https://ac-els-cdn-com.ezproxy.hope.ac.uk/S0268401217308678/1-s2.0-S0268401217308678-main.pdf?\_tid=6dec7999-d8e0-41fa-9aa2-7a00300d8e2b&acdnat=1546748233\_98335223d378434624a7ebdce12aed39. Last accessed 30/12/2018.

Ankit Chaudhary. (2017). *Finger-stylus for non touch-enable systems.* Available: https://ac-els-cdn-com.ezproxy.hope.ac.uk/S1018363914000063/1-s2.0-S1018363914000063-main.pdf?\_tid=a075e6ad-6c43-4e89-9a5f-ecd7b959a89a&acdnat=1547405736\_9447ef303ae2b25b220d8862e5670997. Last accessed 1/1/2019.

Ankit Chaudhary, J.L. Raheja. (2018). *Light invariant real-time robust hand gesture recognition.* Available: https://ac-els-cdn-com.ezproxy.hope.ac.uk/S0030402617315784/1-s2.0-S0030402617315784-main.pdf?\_tid=06c02eb2-ab68-4463-b7c1-eb540644b14d&acdnat=1546750445\_15638c1f2f86be8d4e4f7ee0615361d9. Last accessed 30/12/2018.

Attila Licsár, Tamás Szirányi. (2005). *User-adaptive hand gesture recognition system with interactive training.* Available: https://ac-els-cdn-com.ezproxy.hope.ac.uk/S0262885605001241/1-s2.0-S0262885605001241-main.pdf?\_tid=46228750-0fe9-4edc-b42a-ec30a6e758c2&acdnat=1547154913\_892152fe24554e6c5bb8f741bba02d2f. Last accessed 31/12/2018.

David Rozado, Francisco B. Rodriguez, Pablo Varona. (2012). *Low cost remote gaze gesture recognition in real time.* Available: https://ac-els-cdn-com.ezproxy.hope.ac.uk/S1568494612001159/1-s2.0-S1568494612001159-main.pdf?\_tid=8903a38c-5bcb-4064-96c2-967c98db72e6&acdnat=1546929779\_cb708f630d9e4674498927a69488a020. Last accessed 30/12/2018.

Farid Abedan Kondori, Shahrouz Youseﬁ, Jean-Paul Kouma, Li Liu, Haibo Li. (2015). *Direct hand pose estimation for immersive gestural interaction.* Available: https://ac-els-cdn-com.ezproxy.hope.ac.uk/S0167865515000847/1-s2.0-S0167865515000847-main.pdf?\_tid=99d4197f-d4db-438f-a7c6-c88009d0088a&acdnat=1547248707\_68aa45cf02b5162bcaf6d1718e7d8dfe. Last accessed 1/1/2019.

Feng-Ru Sheu, Nian-Shing Chen. (2014). *Taking a signal: A review of gesture-based computing research in education.* Available: https://ac-els-cdn-com.ezproxy.hope.ac.uk/S0360131514001419/1-s2.0-S0360131514001419-main.pdf?\_tid=1dae4ab4-2f96-44b7-b35d-8c444b641015&acdnat=1546731104\_b6d6852ebba0d80dea01833659f13329. Last accessed 30/12/2018.

Horatiu-Stefan Grif, Trian Turc. (2018). *Human hand gesture based system for mouse cursor control.* Available: https://ac-els-cdn-com.ezproxy.hope.ac.uk/S2351978918304438/1-s2.0-S2351978918304438-main.pdf?\_tid=b87f889e-1c82-4884-9b51-e7305132efc1&acdnat=1547150708\_c6882b0b51a253d92f922fff30d85c5c. Last accessed 31/12/2018.

Hsien-Sheng Hsiao, Jyun-Chen Chen. (2016). *Using a gesture interactive game-based learning approach to improve preschool children's learning performance and motor skills.* Available: https://ac-els-cdn-com.ezproxy.hope.ac.uk/S0360131516300057/1-s2.0-S0360131516300057-main.pdf?\_tid=18fe8c61-ec54-4c22-bf67-5f7baa476b54&acdnat=1546745853\_a1739501d012c2fa4e8801e2d884c1de. Last accessed 30/12/2018.

Sabrina S. Billinghurst, Kim-Phuong L. Vu. (2015). *Touch screen gestures for web browsing tasks.* Available: https://ac-els-cdn-com.ezproxy.hope.ac.uk/S0747563215004513/1-s2.0-S0747563215004513-main.pdf?\_tid=c8825583-e144-4f56-8354-acd346acc565&acdnat=1547243759\_502c6c53482786976d47b5010ed69e7f. Last accessed 1/1/2019.

Shizhi Chen, YingLi Tian, Qingshan Liu, Dimitris N. Metaxas. (2012). *Recognizing expressions from face and body gesture by temporal normalized motion and appearance features.* Available: https://ac-els-cdn-com.ezproxy.hope.ac.uk/S0262885612001023/1-s2.0-S0262885612001023-main.pdf?\_tid=51424a45-e157-4b2d-8dba-fced4af50c86&acdnat=1547061018\_30b1e7914415e7af85e7ed377bcadc1d. Last accessed 31/12/2018.

Stephanie Foehrenbach, Werner A. König, Jens Gerken, Harald Reiterer. (2009). *Tactile feedback enhanced hand gesture interaction at large, high-resolution displays.* Available: https://ac-els-cdn-com.ezproxy.hope.ac.uk/S1045926X09000573/1-s2.0-S1045926X09000573-main.pdf?\_tid=da6a8a16-0650-43f6-a0bc-03b60c56329d&acdnat=1547241848\_4b574f22ce3faf0dc2d3c11fea961e74. Last accessed 1/1/2019.

Zhiquan Feng, Bo Yang, Na Lv, Tao Xu, Jianqin Yin, Xiuyang Zhao, Shichang Feng. (2015). *Motion-towards-each-other-based hand gesture initialization.* Available: https://ac-els-cdn-com.ezproxy.hope.ac.uk/S0031320315001946/1-s2.0-S0031320315001946-main.pdf?\_tid=4a235e20-a447-49cd-ac11-aaac0a39bbec&acdnat=1547055273\_443e165dbdfbe7e710419bfdcd4bc4a3. Last accessed 31/12/2018.

Zhao Lv, Chao Zhang, Bangyan Zhou, Xiangping Gao, Xiaopei Wu. (2018). *Design and implementation of an eye gesture perception system based on electrooculography.* Available: https://ac-els-cdn-com.ezproxy.hope.ac.uk/S0957417417306188/1-s2.0-S0957417417306188-main.pdf?\_tid=946ab09f-ceb1-4a5d-90f5-f584087065a0&acdnat=1547149222\_ac3e3206b26cbbc3a8ce2e7330a45e4b. Last accessed 31/12/2018.

Chapter 7 – Appendices

The appendices contain the HTML, CSS and JavaScript code. It includes all 14 images. It includes the Myo.js Library. All these can be found in the folder accompanying this project. This is all the work required to get the project model itself working.