

Mission Control

Simulation Software

Cathryn Dunicliff

Center number: 30010

Candidate number: 5060

DATE: 16-04-2018

Index

0	0	Index	1 - 3
1	0	Analysis	4 - 16
1		Problem identification	4
2		Proposed solution	4
3		Identification of end users	4
4		Description of current system	5
5		Issues associated with current system	6
6		Further information required to proceed	6
7		Justification of a computational solution	6
8		Identification of processing required to solve problem	6
9		Potential use of decomposition and abstraction to solve problem	7
10		Identification of Stakeholders	7
11		Requirements of Stakeholders	8
12		Realistic review of Stakeholder's current IT skills & effect on solution	9
13		Accessibility requirements for Stakeholders	10
14		Justification of suitability of proposed solution for Stakeholders ***	11
15		Analysis of existing solution: Kerbal Space Program	11
16		Analysis of existing solution: Artemis Spaceship Bridge Simulator	12
17		Measurable Success Criteria	13
18		Identification acceptable limitations	14
19		Objectives for proposed system	15
		Realistic appraisal of the feasibility of potential solutions	15

2 0	Design	16
2 1	Systematic breakdown of the problem	16
	Client & Server Flow charts	17
	Page Functionality Diagram	18
2 2	User interface	19
	Main Ui	19
	Numerical Value Editor	20
	Navball / FDAO	21 - 22
2 3	Usability Features	23
2 4	Security	23 - 24
2 5	System Design	25
	File tree	25
	Justification of choice of external modules	26
	Server Threads and Active Processes	27
2 6	Calculation engine design	28
	Classes	28
	Variable list for resources.py	29 - 30
	Mathematical Functions	31 - 38
	Mathematical Data Flow	39
2 7	Server Design	40
	Variable list for app.py	40 - 43
	Flask app routes	43 - 44
	SocketIO app routes	45
2 8	Client	46
	Functions within the client	46 - 50

2 9	Development Test Plan	51 - 60
2 10	Final Testing Plan	61 - 70
3 0	Development Log	71
	Explanation of Dev log and use of GitHub	71
	Development log & In development Testing	72 - 208
4 0	Final Testing	208 - 224
5 0	Evaluation	224
5 1	Success Criteria	224 - 229
5 2	Usability Features	230
5 3	Maintenance Issues	231
5 4	Limitations and Improvements	232 - 223
6 0	Code As Appendix	234

1.0 Analysis

1.1 Problem Identification

A group of amateur rocketry & avionics enthusiasts need a fun and educational program they can use to test their skills of rocket control, telemetry and mathematical calculations to train for larger missions.

The Flame Trench is a group of individuals particularly interested in avionics and rocketry, including orbital mechanics and both atmospheric and space dynamics. They require a method to test their mathematical and pilot skills to prove they are as good as they need to be before continuing to actual launches, and to identify areas of difficulty they experience.

1.2 Proposed solution

I propose a multiplayer game consisting of a server and multiple clients. On the server is a constantly running and very accurate computer simulation of a rocket launch that the users can directly influence from their remote clients over the internet. The end users would in turn get lots of feedback from the server which could be displayed in graphs and other GUI elements. The game would have lots of different roles, each one representing different jobs that would be required inside a real mission control in order to fly a real rocket. The simulation should be as realistic as possible and the user's display should be as close to a mission control set up as possible.

The different roles would all require the users to communicate with each other to get information to control their part of the rocket. This solution includes accuracy, teamwork, realism and hopefully fun.

1.3 Identification of end users

My end users generally have high IT skills and powerful computers. They are mostly amateur rocket scientists and/or physicists, with a love for gaming.

They are often at university or live far away from their friends so find it difficult to meet face to face. They are often competitive but also enjoy working together to find a solution to a problem. They believe the User interface is a very important element of any software and would be discouraged from playing a game with a bad user interface.

My end users are generally young in their Early-Mid twenties.

1.4 Description of current system

There is currently no available solution that fills the needs and requirements of my client. The only current solution is Kerbal Space Program. The system currently used would be manually calculating data by hand and cross-referencing it against expected values in a Dungeons and Dragons style game. The issue with this is it is incredibly slow and subject to human error, which can make the system a not fun experience. Another issue with this is it's not particularly educational or friendly to newcomers who wish to learn as manually checking your calculations and blindly hoping they're correct is inefficient.

1.5 Issues associated with current system

The current system is slow, and offers no real time feedback. Whilst multiplayer capabilities are possible in a dungeon and dragons esque game, it doesn't offer much in terms of gameplay. The vast amounts of computation that has to be done by hand often take the fun out of playing and it ends up being not fun.

As well as this, the system is subject to human error, often meaning large chunks of the game have to be redone.

1.6 Further information required to proceed

I need to know the general hardware requirements in order to proceed so I can choose an appropriate method and language to develop my solution in.

I also need to know how good at IT my end users are to work out how user friendly I need to make my solution.

In order to design my game, I also need to know what my end user requires of the game and what sort of a person they are. As to how complicated I make the game.

I need to know what kind of processing i'm going to have to do, and what sort of calculations i'm doing which control how accurate the simulation will be, such as whether or not I need to develop a system to calculate n-body orbits.

I need to also know what accessibility features are required, such as colour blind modes and the ability to change font size.

1.7 Justification of a computational solution

Whilst this problem could be solved without aid of a computer program, and has been in the past, the computerised solution offers a far quicker and more reliable simulation to play with. A hand calculated simulation may take days, offering no real time feedback as to how the simulation is unfolding as you go, and is subject to human error. A computerised solution can make a more reliable, and faster way of training for mission control.

1.8 Identification of processing required to solve problem

Processing will be a key part of my program, as there will be a lot of maths, which is better suited to a computer as it is more reliable and a lot quicker than hand calculations. I may also need to process user accounts and profiles which by using a computerised solution eliminates the chance for error.

1.9 Potential use of decomposition and abstraction to solve problem

My proposed solution could end up being incredibly complicated if I model every object and event in my simulation. I plan on abstracting this problem by only simulating key components that make an obvious difference to the game and to the maths.

I also plan on abstracting and decomposing the problem of programming tasks into functions and procedures so I don't have to solve big problems in one go but can instead break it down into smaller, easier to handle problems. This also means certain functions can be reused saving time and processing power later on.

1.10 Identification of Stakeholders

My stakeholders are from **The Flame Trench**.

A group of British and American hobbyist engineers in the rocketry/space industry.

My main Stakeholders are listed below, however all members will be involved at some point in the development of the proposed solution by answering surveys and giving feedback via an online IRC channel.

Jonny Hyman 20 year old Engineering & Design student in Los Angeles, CA.
Interested in aviation & physics.

Very competent with computers.
Uses Windows and Linux



William Easdown 20 year old student of Integrated Mechanical & Electrical Engineering at university of Bath.
Interested in amateur rocketry.
President of SpaceSoc at Bath.

Quite competent with computers.
Uses Windows



Ben Cartwright 21 year old Student/spacecraft electronics engineer at Aston university. Interested in amateur rocketry.

Very competent with computers.
Uses Windows



Other members include:

- Andy Esser
- Arsenio Menendez
- Carol Wright
- Joe Barnard
- Sarah Averill

1.11 Requirements of Stakeholders

At the start of research, a questionnaire was presented to my stakeholders. The following information is based upon the feedback.

A full copy of the original questionnaire and responses can be found at the back of this report.

- Must:
- Be multiplayer
 - Have accurate physics simulations
 - Allocate each connected computer a different role in mission control
 - Allow the host to control aspects of the game
 - Allow users to interact with the game and change the course of events
 - Have a GUI

- Should:
- Have a nice user interface
 - Allow games to be password protected
 - Save data locally

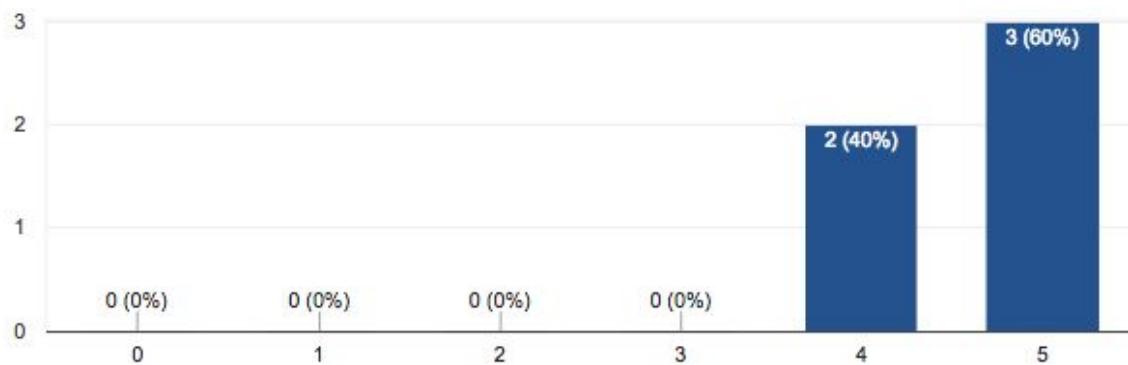
- Could:
- Include a tutorial mode
 - Include pre-built scenarios
 - Allow different modes of difficulty
 - Have day/night themes

1.12 Realistic review of Stakeholder's current It skills & effect on solution

From my questionnaire, I found that all my stakeholders are quite to very competent with computers.

How competent would you say you are with computers?

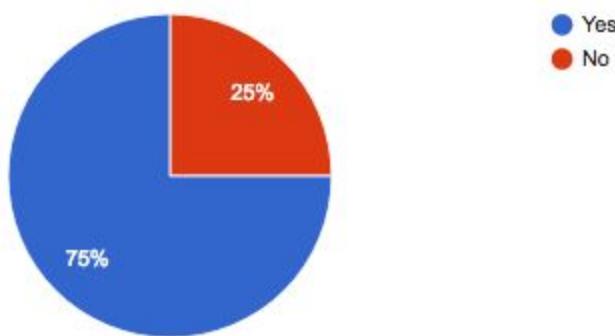
5 responses



This means that my user interface doesn't need to be too simplified and I can make assumptions that the user already knows how to use the user interface.

Would you be comfortable running the game from the command line?

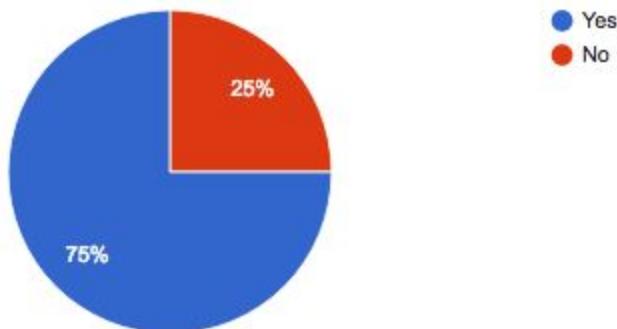
4 responses



I also found the majority of people would be comfortable running the game from the command line, meaning I don't have to make a compiled program and can likely just distribute the source code. This also shows that most people are familiar / confident enough with the command line should I need to utilize it.

Would you be comfortable setting up your own server to host the game?

4 responses

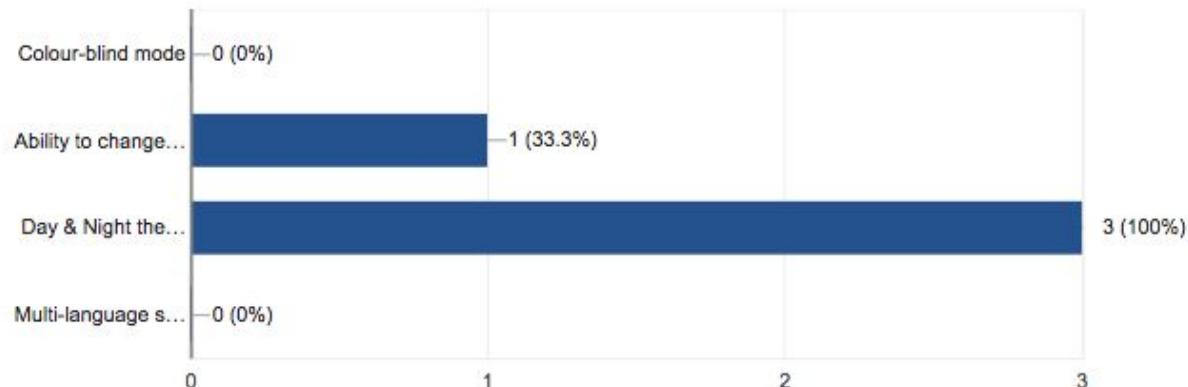


My results show that the majority of people would be happy running their own server rather than using online hosting meaning I don't need to make a service that can run independently to host games and can produce a package that a user can host and control themselves.

1.13 Accessibility requirements for Stakeholders

What accessibility features are important to you?

3 responses



My results show that 100% of people who answered would like a day or night mode, so I need to try include a workaround for this in my solution.

1.14 Justification of suitability of proposed solution for Stakeholders

The proposed solution is suitable for my stakeholders because it matches their required criteria. The solution proposes a multiplayer game which would be what they'd like, and it allows openings for team work within the game. The solution also would be incredibly accurate and fast compared to hand calculating and checking maths, which would be ideal for training and fine tuning rocket launch flight controls and mathematical capabilities.

1.15 Analysis of existing solution: Kerbal Space Program

Kerbal Space Program

Kerbal space program is a single player rocket simulation game in which you can build and launch multiple rockets. It has a detailed graphical user interface that is simple to use, and uses complex mathematical simulations to create a high accuracy simulation. It currently has no capacity for multiplayer games, however is a good introduction to orbital mechanics.



The user interface is purely graphical giving a good representation of what's happening in the game and how the changes you make impact the simulation.

This game isn't a suitable solution for my stakeholders' because it lacks multiplayer functionality and higher level mathematical capabilities that would be needed to test their skills as requested.

Advantages:

- Very accurate physics emulation
- Easy setup
- Lot's of customizability
- Visual simulation enhances learning

Disadvantages:

- Single player only (without modification)
- A visual interface distracts from the maths behind the simulation
- CPU & GPU intensive - won't run on a chromebook
- Uses N-body not 2-body orbital mechanics

Features I can implement into my program:

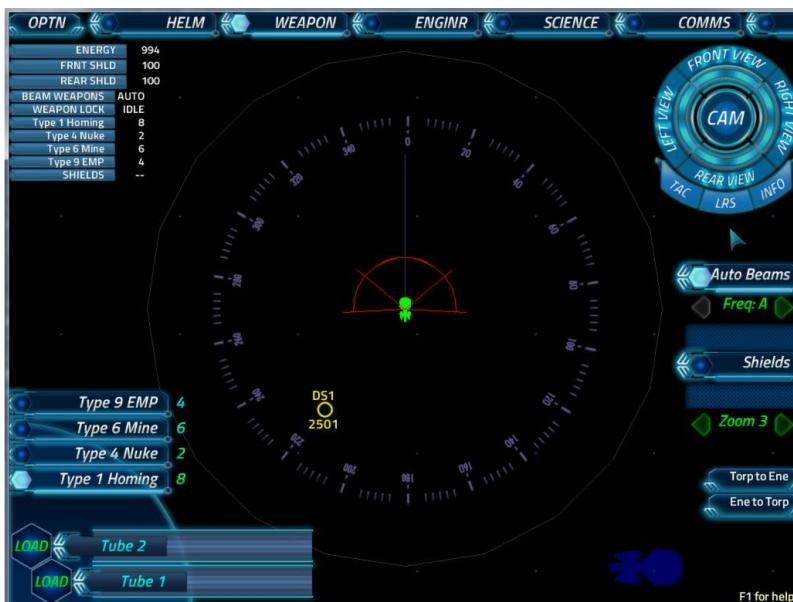
- Accurate physics simulations
- Easy setup
- Some GUI elements can be used in my UI (navball etc)

1.16 Analysis of existing solution: Artemis Spaceship Bridge Simulator

Artemis Spaceship Bridge Simulator

Artemis Spaceship Bridge Simulator is a multiplayer game in which players are positioned in different roles on the bridge of a spaceship. Each role serves a key function to the successful flight. It is focused primarily on teamwork and communication, requiring users to talk to each other frequently and efficiently. The game features space battles and side quests to make the experience more interesting, however doesn't feature accurate physics simulations or even rocket launches, meaning it isn't currently meeting my stakeholders' needs.

Artemis features a full Graphical User Interface that is easy to use and looks great:



The GUI is simplistic and clear which is something that I know from my research that my stakeholders' value.

Advantages:

- Forces teamwork and communication
- Multiplayer
- Easy to understand and use
- Graphical User Interface
- Gives each player a different role
- Includes pre-built scenarios to maintain interest

-
- Disadvantages:
- Doesn't have accurate physics simulations
 - Doesn't model rocket launches and flights
 - Doesn't test mathematical and piloting skills

- Features I can implement into my program:
- Multiplayer capabilities
 - Easy to understand Interfaces
 - Clear GUI
 - Each player can choose a different role.

1.17 Measurable success criteria

Success Criteria 1: My system must be networked

- The game must be accessible by a browser
- The game must be multiplayer
- The system must have a server
- The system must work in real time

Success Criteria 2: My system must have accurate physics simulations

- There must be Aerodynamic drag simulations
- There must be Gravitational simulations
- There must be orbital mechanical simulations
- There must be standard rocket calculations
- It must feature 3D vector coordinates
- It must be able to calculate results iteratively in real time

Success Criteria 3: My game must have a user interface

- The game must feature a customisable user interface
- The UI must be intuitive and familiar
- The UI must allow for different tools to be used simultaneously
- The UI must have an organised way of navigating through different tools
- The UI must have some visual representations of the game data

Success Criteria 4: My system must be easy to host

- The server must feature an admin run “wizard”
- The server must have live feedback of what’s happening

Success Criteria 5: My system must have game elements

- The game must have an achievable objective
- The game must have an element of user input that directly affects the course of events
- The game must have event scenarios
- The game should include extra non-flight aspects such as life support

1.18 Identification of acceptable limitations

My plans for the project is to build a system that can expand easily and have new functions added quickly. The main structure for the server and client, and the mathematical side of the simulation must all be in place and ready to add more functionality later. Because of time there will be significant limitations that won’t be included in my final version, however could probably be added easily after my project is finished.

Limitations

- The User interface will be purely functional and not particularly aesthetically pleasing. This is mostly due to time constraints. It would be better to get a working product then go back after and make the user interface look nice than to focus on the user interface at the start and end up with a non-functional but aesthetically pleasing product.
- The system will not feature much user management / admin features. I plan on focusing purely on the accuracy of the calculations and the mathematical simulation, as well as the server and clients. There is no point having admin features or user management for a system that provides no other function. I have made this decision due to time constraints limiting the amount of work I can do.
- The computations will not be perfect and will be heavily abstracted. I plan on designing and focusing on the key elements of the simulation that will make a large impact on the flight of a rocket (air resistance, gravity, thrust etc). I will not focus on elements such as wind, aerodynamic heating, fuel flow dynamics etc. This limit is again down to time. As well as this there is a limit to how accurate the results of the simulations can be. A computer can only store a certain number of significant figures and when dealing with

irrational numbers like Pi, it means that the numbers will always be an incredibly small fraction off. E.g the rocket will lose a meter every 100 million meters.

- The user interface and server will only have a small number of tools and functions. E.g. In the user interface there may only be the option to change the speed and direction of the rocket. It is unlikely that due to time I will manage much more. My primary focus is on getting a versatile and expandable minimum viable product ready.

1.19 Objectives for proposed system

- Game will include a server that the host/admin can set up
- The users can connect to the game remotely in a multiplayer setup
- The users will be presented with various different tools and controls that send data to the server that affect a simulation
- The server simulates a rocket launch and sends the data to the users
- The users will have a clear objective in the game that they must succeed in to win
- The game must require users to rely on their mathematical skills (orbital mechanics etc)

1.20 Realistic appraisal of the feasibility of potential solution

I believe the solution is feasible to a degree. A server and client setup is definitely feasible and a basic simulation server side would also be feasible however due to time limitations I would be concerned about how many game mechanics and fun features I could develop and implement in that time. I believe a minimum viable product with a proper framework would be the correct aim for this project, as it can be expanded and built upon easily if the correct framework is in place after the time is up if need be. The correct framework would include a server and client communicating with a game interface and simulation.

I also worry, however, that by only aiming for a minimum viable product to start with and building from there, that there won't be much game aspect to the game. My stakeholders have specifically requested a multiplayer game and whilst the game will be multiplayer capable it likely wouldn't require more than one player.

If time allows it would be good to put small game mechanics in that require multiple people to operate.

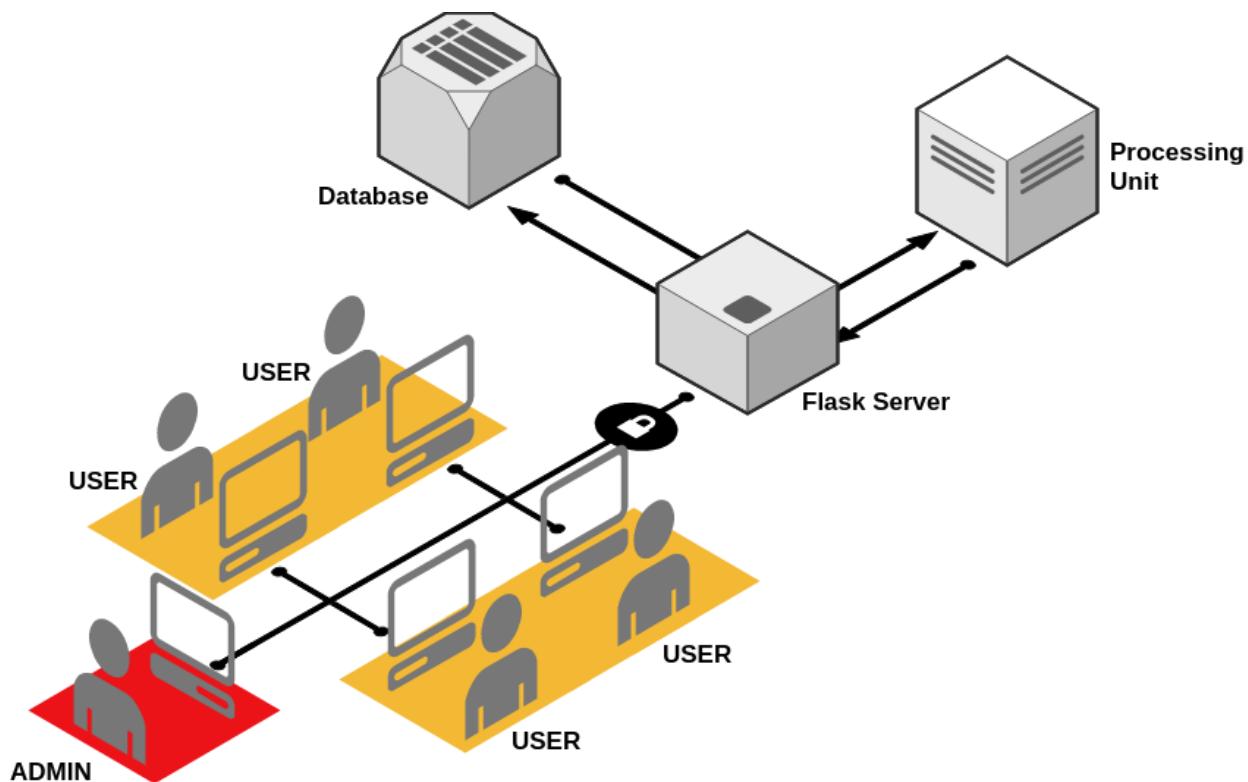
In conclusion I believe this project is certainly feasible however the limitations listed above will apply heavily.

Design

2.1 Systematic Breakdown of the problem

Simplified Initial Model

The below is a very simplified version of the game showing how the users will interact with the server and each other.



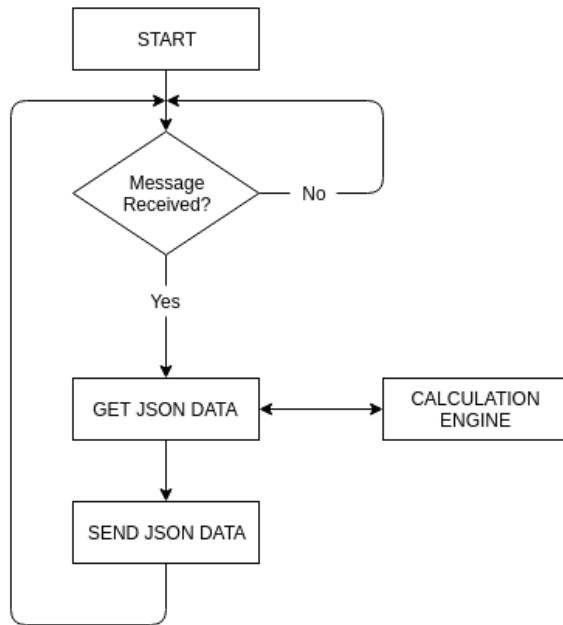
This diagram assumes all the users are connected on the LAN however if necessary port forwarding could be used to play over WAN.

Simplified Client and Server Flow Charts

These flow charts are my initial ideas of how the server will send game data to and from the client. The data could be anything from a list of players to complex rocket launch data.

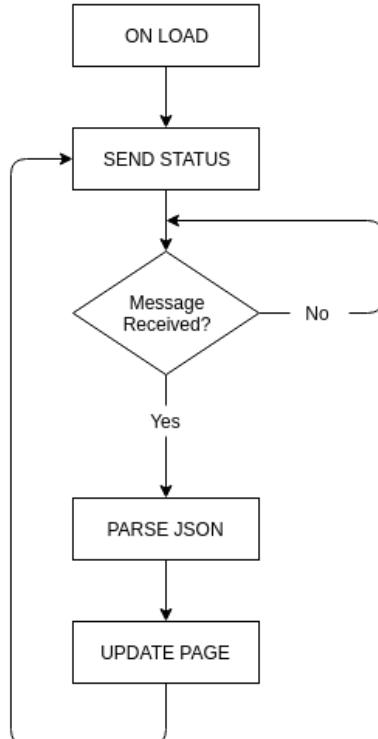
Server

The server mechanism should simply wait until a message is received from the client over web-sockets. When a message is received it should get/run the latest calculations and send them to the client.



Client

The client follows a very similar system. It sends a message when it connects to the server. It then waits for the JSON data. When the data is received it will parse and update the page with the new data. The process repeats.



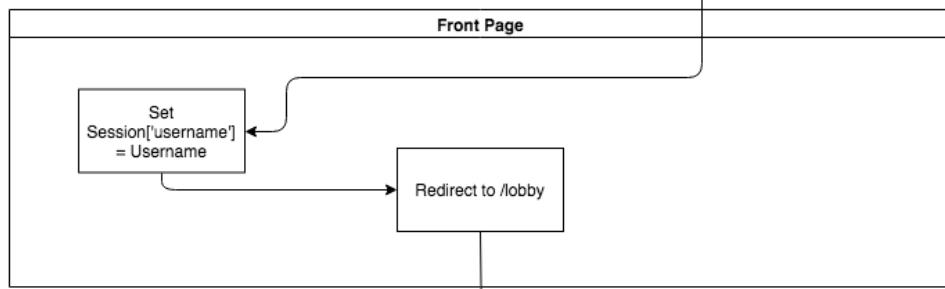
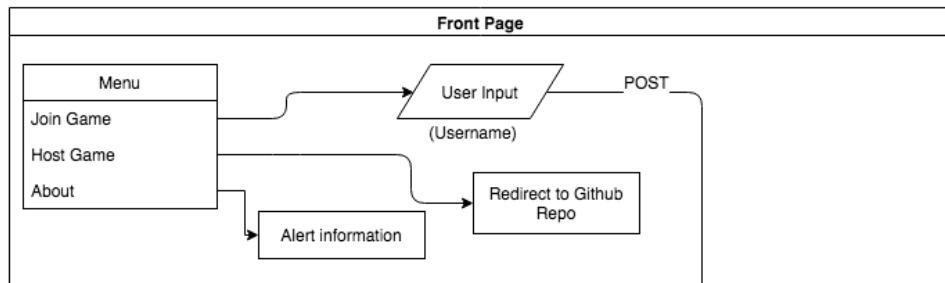
Front Page

The front page will greet the user and give them the option to join or host a game, or get more information about the game. Should the user choose join, it will request a username that will be sent via web-sockets to the server. Selecting host will give you instructions on how to download and install the host on the project GitHub repo. About will give you a basic alert message telling you about the game.

Login

This page receives the POSTed username from the front page and passes it on to the server as the user name so it can set it as session ID and then redirects the client to /lobby.

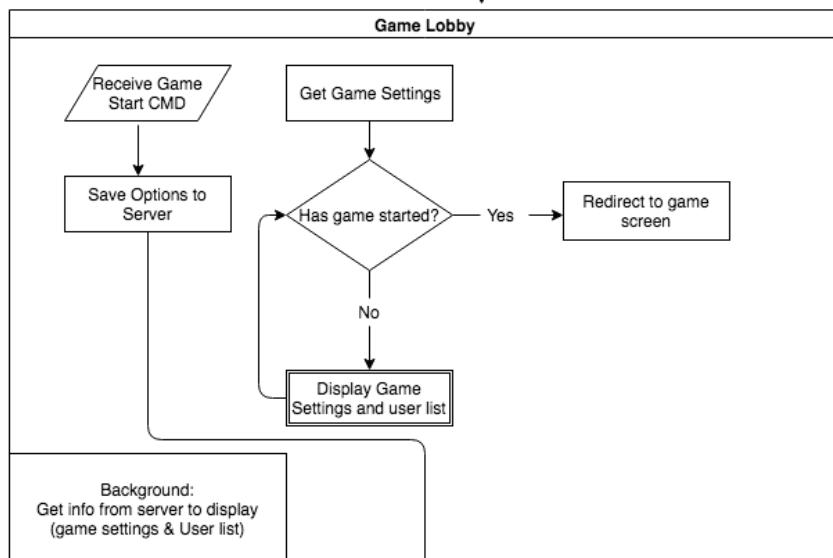
This is not a HTML page but is a server process



Game lobby

This is where the user will wait until the game starts. It should display any relevant info i.e settings, other players connected. It should give the user game options that the administrator has allowed in game permissions.

It should also check to see if a game has already started, and if so it should automatically redirect to the game screen

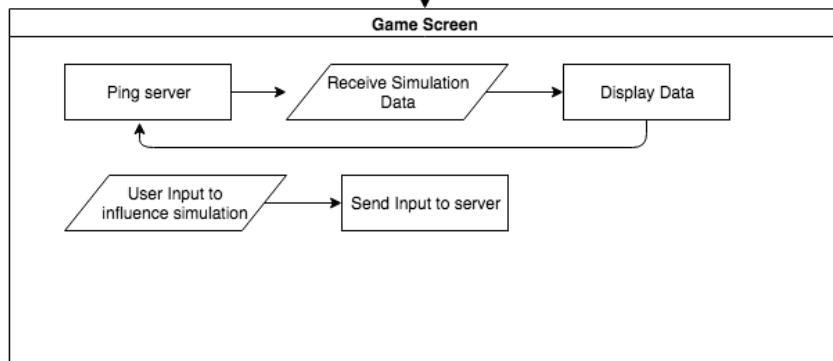


Game screen

This is the main game screen the user will use.

It will constantly update any simulation data by fetching it from the server.

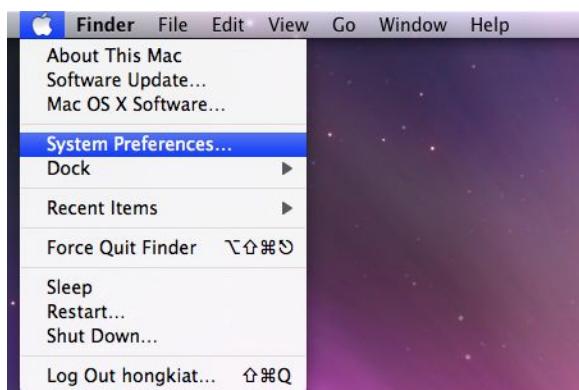
It will also take user input and send it to the server to be fed into the simulation to react.



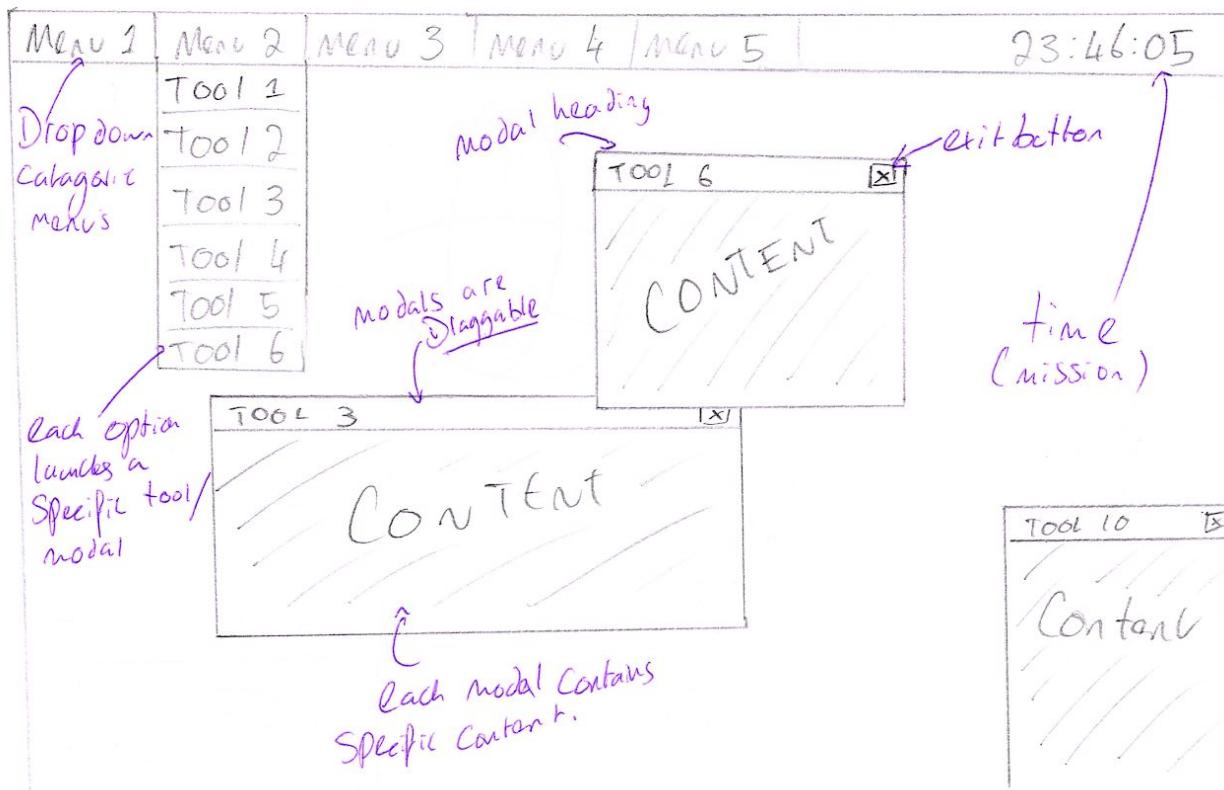
2.2 User interface

Main UI

This is the main game screen where the user will spend most of their time. It must contain all the tools and visuals within the game that the user directly interacts with. To contain this vast amount of information and controls in a simple and organised way, it needs to have a layout that requires little to no explanation. A familiar interface would be a good start, but it must be minimal to not take over the screen.

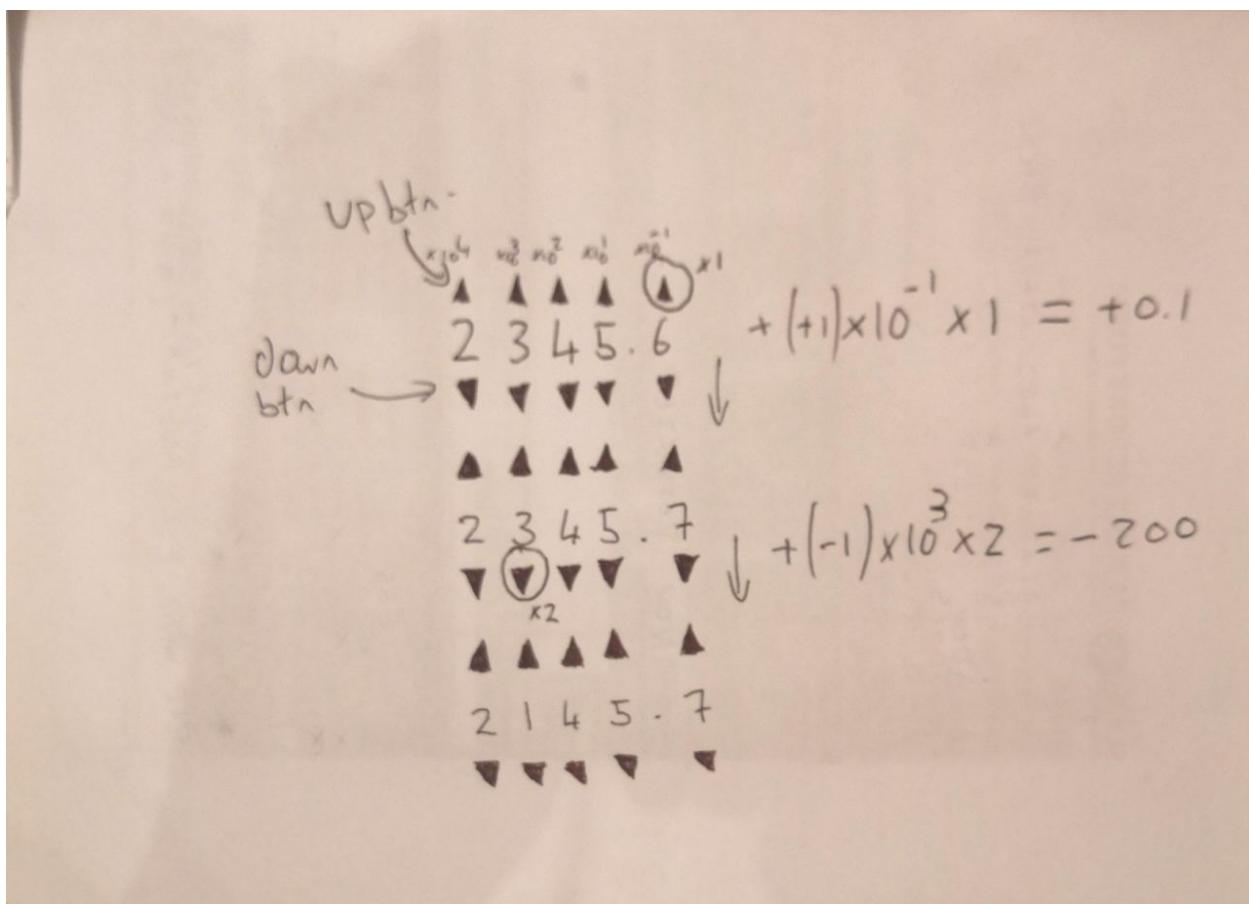


One option is to use a top navigation bar, familiarly seen in Mac and Linux operating systems. The design is minimal, taking up very little space but is expandable when required, meaning it can fit a lot of content inside. I think this would be an ideal choice for my UI navigation, however some users may need some introduction, although I feel it's pretty self explanatory.



The above is my Final design. It features a drop down navigation bar along the top that can contain buttons that expand modals with content in. The design is elegant and leaves lots of room for the draggable content boxes (modals). Also along the navigation bar will be critical information, such as the mission time, and perhaps a warning light. The modals will follow standard window design style to make sure the users are familiar with the interface and it requires no explanation. Along the top of the modal will be the content heading, and an exit button. The modals should be draggable but only by this point (a handle). I think it would be good if the modals, when dragged, went slightly transparent to allow users to see behind when positioning. This method of allowing users to move the content as they see fit creates a custom workspace for every user, helping them get the most of the game.

Numerical Value Editor



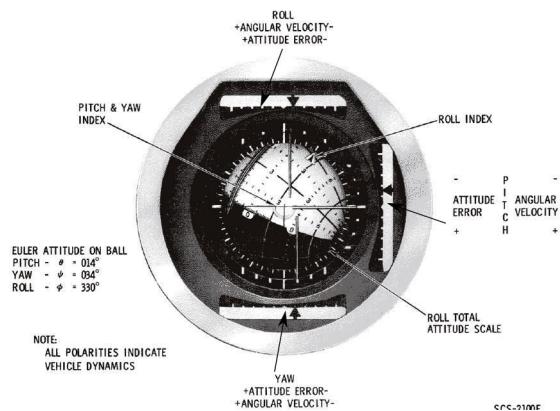
The above design is my way of simplifying user input for precise values. Rather than just a text input field where they can type in a float value, this design allows my users to simply change 1

significant figure at a time by pressing up and down arrows. It also allows them to see the number they change so they don't forget it. It works by pressing the up or down arrow above the significant figure you wish to change and that number will move up or down by 1.

So the first up arrow on the number 2000 would ADD(up arrow) 1E4 as the first up arrow would be the 4th number from decimal point. So $2000 + 1E4 = 3000$, hence the 2 would change to a 3 and the new value would be 3000.

I believe this is an excellent design for my users and will increase usability dramatically.

Navball / Flight Director Attitude Indicator (FDI)



The Navball shows three things at once. The total attitude, the attitude rate and the attitude error relative to a previously defined other attitude.

This will show the user the Euler angle coordinates indicating the relative angle of the rocket, but not direction of travel.

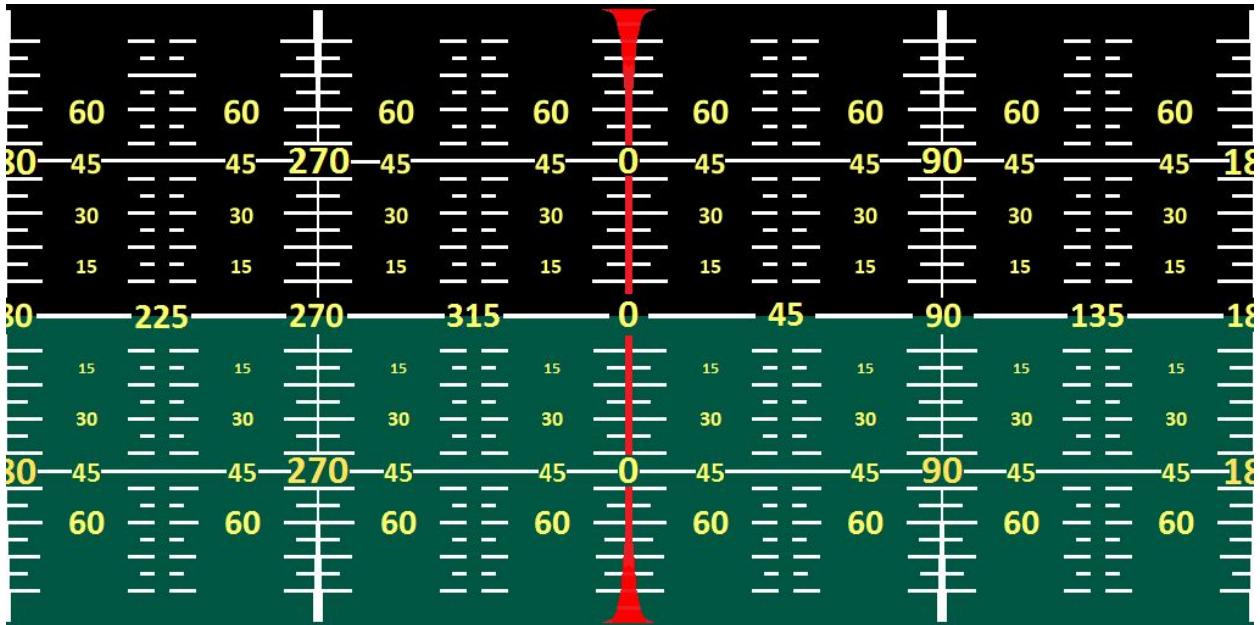


The two images above are already existing designs of navballs. The left a reconstruction of an actual FDAO onboard the Apollo CMS, and the right a navball from the popular rocket simulation game *Kerbal Space Program*.

My navball should follow these designs closely whilst also being simple and incredibly clear to read.

Given this UI aspect will have to update in real time it will likely require animation. The easiest and most lightweight way to do this is via three.js, a WebGL javascript 3D engine. Processing for this animation is done on the client side in browser.

I have designed this map for my navball which will wrap around a sphere:



I asked my stakeholders if they prefer a separate top and bottom and they all said they would definitely prefer separate colours for top and bottom of the Navball. The text is clear and the colours feel sci-fi enough whilst still being bold and readable.

2.3 Usability features

- Tools are nested inside modals (windows) that use a similar mechanic to operating system windows. This makes them easy to know how to use for even the least computer literate users. It is draggable like a normal OS GUI window to allow users to arrange their workspace as they wish. The exit button is conveniently in the top right which is where Windows users (the majority of modern computer users) are used to it being.
- Tools are organised and can be open by the navigation bar, another familiar Operating System GUI feature and also often used in websites meaning the user will be familiar with it so make my game easier to use. The drop down system allows for tools to be within categories. This is important to help organise the tools when the software has a lot. The navbar with categories and tools will be familiar in packages like Microsoft Word.
- Easy Numerical Value changers. My program will feature a way of easily changing a specific digit of a number by a power of 10 so you don't have to type out the number quickly. It is very intuitive to use. I have included this rather than a standard text entry box because it gives live feedback and is easier to use on the fly.

2.4 Security

Since my project is multiplayer and utilises the internet I have to consider what security measures need to be in place to protect the server and the user.

Server - The server is run by the user locally. I cannot do anything to make their Local network more secure. If the user decides to port forward to open the application to a Wide area network they will have to provide their own security measures to prevent unlawful access.

Databases - my application doesn't use a database so isn't subject to database leaks or SQL Injection.

Data injection - my application uses websockets. If someone was on the same network and had the correct websocket protocols they could read or inject data into the application. This is important to stop as in a competitive scenario it could sabotage one group's game. To eliminate this problem the websockets use a 'secret key' which is used for encryption of websocket data. I will make this definable in the CONFIG.txt file (server settings). This secret key also illuminates python code injection as special characters like ', & ; etc will be changed to other characters. If

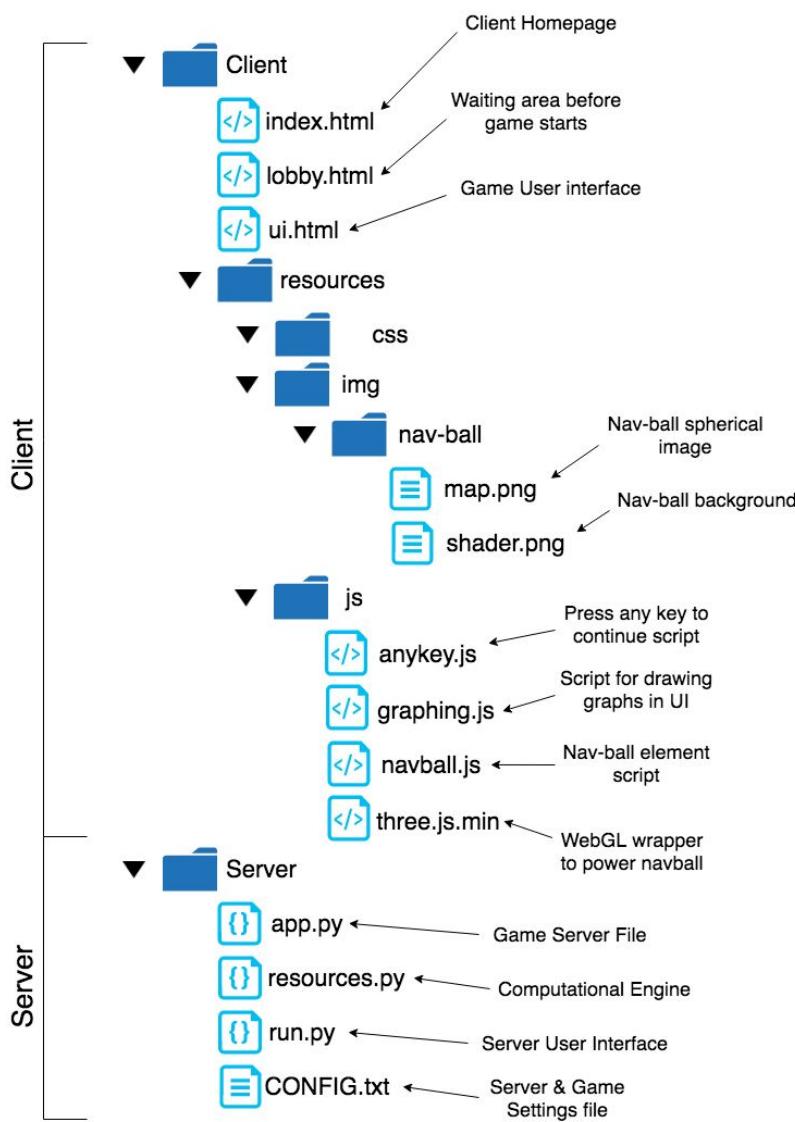
not Flask automatically converts them to their unicode codes.

Cheating - if you are a user playing the game you could very easily use chrome's javascript console (or similar) to send correctly formatted data to the server to change the position of a rocket into a desirable position (orbit) or change the fuel remaining etc. The information on how to do this would be publicly accessible in the client side source code, which I have no control over. To reduce damage from cheating, I will reduce the amount different variables the client can change. E.g. They will be able to change Thrust, angle (input variables) but will not be able to control variables produced by the computation engine such as position, velocity etc.

2.5 System Design

File tree

Below you can find my program's file tree. This can be split into two main parts, the server and the client. Since the server delivers the client pages to the user via HTTP they cannot be split without a large reworking of the server code. Inside the client folder, different elements such as Javascript code and images are split further. I have done this as it is important to keep an organised file system, both for debugging / development purposes and to easily find files and to identify problems. When coding it makes creating paths a lot easier since files of a certain type will always be found within the same directory.

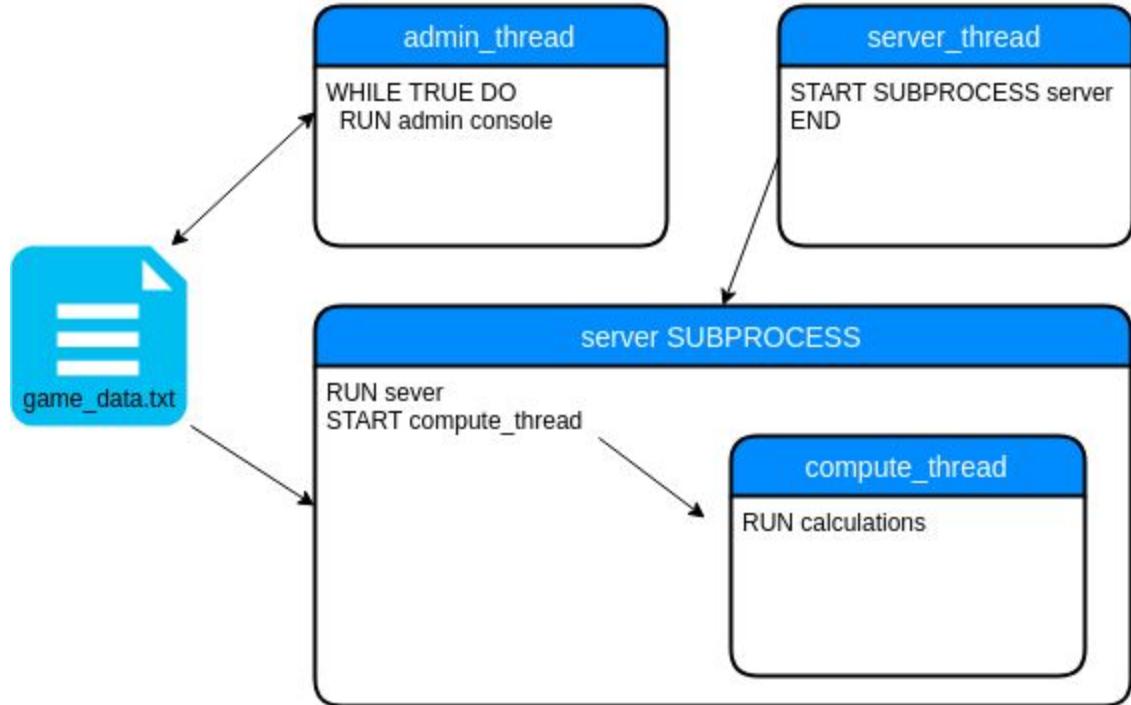


Justification of choice of external modules

Module Name	Description	Where it will be used	Justification
Numpy	A python module used for complex mathematical programming.	Computation engine to calculate game data before being passed to server.	Numpy will calculate to a much higher degree of accuracy than normal python's math module. It also includes more complex mathematical functions required for my game and better data structures than python's arrays.
Flask	A python module for hosting servers in python.	In my server to host the game content and provide responses to clients.	It's important to have a server for my project to allow the multiplayer aspect to work. I'm using a python module rather than something like Apache or NGINX for continuity and to make it easier to do my computations. I'm using Flask because it has the most features for any python server module
Flask-SocketIO	A Flask bolt-on python module that allows for websocket communications.	In the server to receive and send important game data.	SocketIO is the only websocket module for python. I have chosen to use web-sockets because of their optional security, speed and compatibility with modern browsers. The module makes it a lot easier to send/receive data than by using raw sockets.
SocketIO JS	A javascript client-side websocket communication module.	It be used on my client pages to receive game data ready to display it, and to send instructions to the server.	This is a very lightweight module for clients, and requires the client to have nothing installed first as it can be served by the

			server or by CDN. It works in conjunction with the SocketIO server module.
--	--	--	--

Server Threads and Active Processes



Name	Type	Description	Time Active
admin_thread	Thread	This thread runs the admin console which controls the server and game in run.py	Until quit
server_thread	Thread	This thread starts the server subprocess then dies.	1 Instruction
server	Subprocess	This subprocess is the entire server for the game.	Forever (until stopped by server)
compute_thread	Thread	This thread within the	Whilst game is active

		server subprocess runs the server's calculations.	
--	--	---	--

2.6 Calculation Engine Design

Classes

Name	Inherits	Methods	Attribute	Other
GameObject()		Constructor() Distance() Gravity() Get_grav_vector()	Mass, Radius, Position, Velocity, Acceleration	
Planet()	GameObject()	Constructor()	Atmospheric pressure at sea level, Atmosphere's molar mass	class Atmosphere()
Atmosphere()	Planet()	Constructor() Pressure()		
Rocket()	GameObject()	Constructor() Resolve_thrust() Thrust() Drag_area() Drag()	Length, Radius, F, Angle, Rate of ejected mass flow, Exhaust gas ejection speed, Exhaust gas pressure, Area of gas exit, Coefficient of drag, Flow velocity of gas	

Variable list for Resources.py

Name	Type	Location	Description
OBJECTS	List	Global	A list of all the game objects instantiated
data	Dictionary	Global	A dictionary of variables to be passed to the client
p_acc	Array[3]	data	Player's acceleration
p_vel	Array[3]	data	Players' velocity
p_pos	Array[3]	data	Player's position
p_ang	Array[3]	data	Player's angle
p_fue	Float	data	Player's fuel
p_thm	Float	data	Player's thrust
p_sta	Boolean	data	Player's status
p_orb	Boolean	data	Player's orbital status
mass	Float	class GameObject	Object's mass
radius	Float	class GameObject	Object's radius
pos	Array[3]	class GameObject	Object's position
V	Array[3]	class GameObject	Object's velocity
a	Array[3]	class GameObject	Object's acceleration
p0	Float	class Planet	Atmospheric pressure at sea level
molMass	Float	class Planet	Molar mass of atmosphere
length	Float	class Rocket	Rocket length
F	Array[3]	class Rocket	Force on rocket

angle	Array[3]	class Rocket	Rocket angle
q	Float	class Rocket	Rate of ejected mass flow
Ve	Float	class Rocket	Exhaust gas ejection speed
Pe	Float	class Rocket	Pressure of exhaust gasses
Cd	Float	class Rocket	Rocket coefficient of drag
Vg	Float	class Rocket	Velocity of ejected gas
Ae	Float	class Rocket	Area of fuel exit
Fuel	Float	class Rocket	Fuel left in rocket

Mathematical Functions

Name	Parameters	Explanation
distance	Object1.position, Object2.position	<p>This function can be used to find the scalar distance between any two objects using a modified version of pythagoras' theorem</p> $r = \sqrt{ i_1 - i_2 ^2 + j_1 - j_2 ^2 + k_1 - k_2 ^2}$
gravity	Object1.mass, Object2.mass, Object1.position, Object2.position	<p>This function takes the mass and positions of 2 objects to calculate the gravitational attraction between the two. This will be done iteratively for every object in the game using the formula:</p> $F = \frac{GMm}{r^2}$ <p>Where G is the gravitational constant 6.67e-11, M is the mass of the object creating the field, m is the mass of the object in the field (that F is applied to), and r is the radius between the two objects.</p> <p><u>Pseudocode</u></p> <pre>Function gravity(m1, m2, r): F = (6.67408e-11 * m1 * m2) / (r**2) return F</pre>
get_grav_vec otr	Scalar force due to gravity, Position of rocket, Distance to object causing field	<p>This function takes the scalar gravity and makes it a vector force using the equation:</p> $F_x = F \frac{i}{r}, F_y = F \frac{j}{r}, F_z = F \frac{k}{r}$ <p>Where Fx, Fy, Fz are the gravitational force vectors, i, j, k are the position of the rocket, and r is the distance between the two object's center of masses.</p> <p><u>Pseudocode</u></p> <pre>Function get_grav_vector(F, i, j, k, r): fx = i/r*F fy = j/r*F fz = k/r*F return [fx, fy, fz]</pre>

pressure	Temperature, Height from sea level, Pressure at sea level, Molar Mass of atmosphere, Local gravitational acceleration, planetary mass	<p>This function calculates the atmospheric pressure exerted on an object at a specific point above a planet's surface. This will be done to aid other functions such as the drag function and thrust function. It utilizes the equation:</p> $P = P_0 e^{(-\frac{Mg}{RT}h)}$ <p>Where P is the atmospheric pressure, P_0 is the pressure at sea level, M is the molar mass of the atmosphere, G is the force due to gravitational acceleration, R is the universal gas constant, T is the ambient atmospheric temperature, and h is the height of the rocket above sea level.</p> <p><u>Pseudocode</u></p> <pre>Function pressure(p0, molM, T, h, g): P = p0*exp((-1*(molM*g) / (8.3145*T)) *h) return P</pre>
thrust	Rate of ejected mass flow of rocket, Exhaust gas ejection speed, Pressure of exhaust gasses, Pressure of ambient atmosphere	<p>Calculates the force generated by a rocket's engines each cycle using the formula:</p> $F = qV_e + A_e(P_e - P_a)$ <p>Where F is the thrust force, q is the rate of ejected mass flow, V_e is the exhaust gas ejection speed, A_e is the area of gas exit, P_e is the pressure of exhaust gases, and P_a is the pressure of the ambient atmosphere.</p> <p><u>Pseudocode</u></p> <pre>Function thrust(q, Ve, Pe, Pa, Ae): F = q*Ve + (Pe - Pa) *Ae return F</pre>
resolve_thrust	Scalar Thrust, Rocket Euler angle	<p>This takes scalar thrust force F and makes it a 3D vector force using the equation:</p> $F_x = F \cos(\alpha_y) \cos(\alpha_p)$ $F_y = F \sin(\alpha_p)$ $F_z = F \sin(\alpha_y) \cos(\alpha_p)$ <p>Where $F(x, y, z)$ are the vector thrusts, and $\alpha(y, p)$ are the yaw and pitch angles.</p>

		<p><u>Pseudocode</u></p> <pre>Function resolve_thrust(F, ap, ay): Fx = F*cos(ay)*cos(ap) Fy = F*sin(ap) Fz = F*sin(ay)*cos(ap) return [Fx, Fy, Fz]</pre>
drag	Atmospheric pressure, Rocket speed, Coefficient of drag of rocket, Area in drag of rocket, Ambient temperature	<p>Calculates the atmospheric drag on a rocket at a given speed and height. This will be done three times for the x, y, and z vectors, and run at every cycle using the formula:</p> $F = \frac{1}{2} \left(\frac{\rho}{RT} u^2 C_d A \right)$ <p>Where F is the resultant drag force, ρ is the atmospheric pressure (PA), R is the air gas constant (286) T is the ambient air temperature, u is the flow velocity of the gas relative to the rocket (rocket speed), C_d is the coefficient of drag of the rocket, and A is the area in drag.</p> <p><u>Pseudocode</u></p> <pre>Function drag(P, V, C, A, T): F = 0.5 * ((P / (286 * T)) * (V**2) * C * A) return F</pre>
drag_area	Rocket height, Rocket radius, Rocket velocity,	Calculates the area in drag of the rocket at any given angle and speed, to be used in aid of the drag function. It creates a scalar area of drag to be passed into the

	<p>Rocket Eulerian angle expressed as 3D vector pointer</p> <p>drag() function, which can then later be resolved, and requires the angles to be resolved to an x, y, z pointer vector rather than eulerian angles. This uses the equation:</p> $ A = rh * \sin(\cos^{-1}\left(\frac{v_x a_x + v_y a_y + v_z a_z}{(\sqrt{v_x^2 + v_y^2 + v_z^2})(\sqrt{a_x^2 + a_y^2 + a_z^2})}\right))$ <p>Where A is the scalar area in drag, r is the radius of the rocket, h is the height of the rocket, V_x, V_y, V_z are the vector velocities of the rocket, $\alpha_x, \alpha_y, \alpha_z$ are the vector angles in the form of a vector pointer</p> <p><u>Pseudocode</u></p> <pre>Function drag_area(V, a, r, h): A = r*h Top = arcos((V[0]*a[0]+V[1]*a[1]+V[2]*a[2])) Bottom = ((sqrt((a[0]**2)+(a[1]**2)+(a[2]**2)))* sqrt((V[0]**2)+(V[1]**2)+(V[2]**2))) A = A * (Top/Bottom) Return A</pre>
--	---

orbit_check	Rocket Velocity, Distance from planet, Planet Mass, Height of edge of atmosphere	<p>This function calculates if a rocket (or other body) is in orbit of a planet or massive body, to be used as a game objective.</p> <p>In the equations below, if R_p (orbit periapsis) > height of edge of atmosphere, then the rocket is in a successful orbit.</p> <p>It uses the equation below:</p> $\gamma = \cos^{-1}\left(\frac{v_x r_x + v_y r_y + v_z r_z}{\sqrt{v_x^2 + v_y^2 + v_z^2 + \sqrt{r_x^2 + r_y^2 + r_z^2}}}\right)$ $R_p = \frac{\sqrt{r^2(G^2 M^2 - 4G \sin^2(\gamma) + 2\sin^2(\gamma) r v^2) + GM r}}{2GM - rv^2}$ <p>Where R_p is the orbital periapsis, V_x, V_y, V_z are the vector velocities of the rocket, r_x, r_y, r_z are the vector radius from the center of the planet, G is the gravitational constant, M is the mass of the planet, r is the scalar radius</p> <p>It is worth noting that the equation above doesn't account for the rocket's mass. This is because in most situations it is so small that it won't make much of a difference, so this can be somewhat abstracted.</p> <p>It is unlikely that an orbit check would ever be performed on two massive bodies where one isn't the rocket.</p> <p><u>Pseudocode</u></p> <pre> Function orbit_check(v, r, m, h): y = (v[0]*r[0]+v[1]*r[1]+v[2]*r[2]) y = y/(resolve(v)+resolve(r)) p = resolve(r)**2*(G**2*M**2-4G*sin(y)**2+ 2sin(y)**2*resolve(r)*v**2)+G*M*resolve (r) p = sqrt(p)/2*G*M-resolve(r)*v**2 if p > h: return True else: return False </pre> <p>Here I have used the resolve() function to indicate the resolution of a 3-part vector to a scalar using</p>
-------------	---	---

		pythagoras as seen previously. I have written it here as a function to show it will be written as a function and used in my equations as so, because of the amount of times it's used.
--	--	--

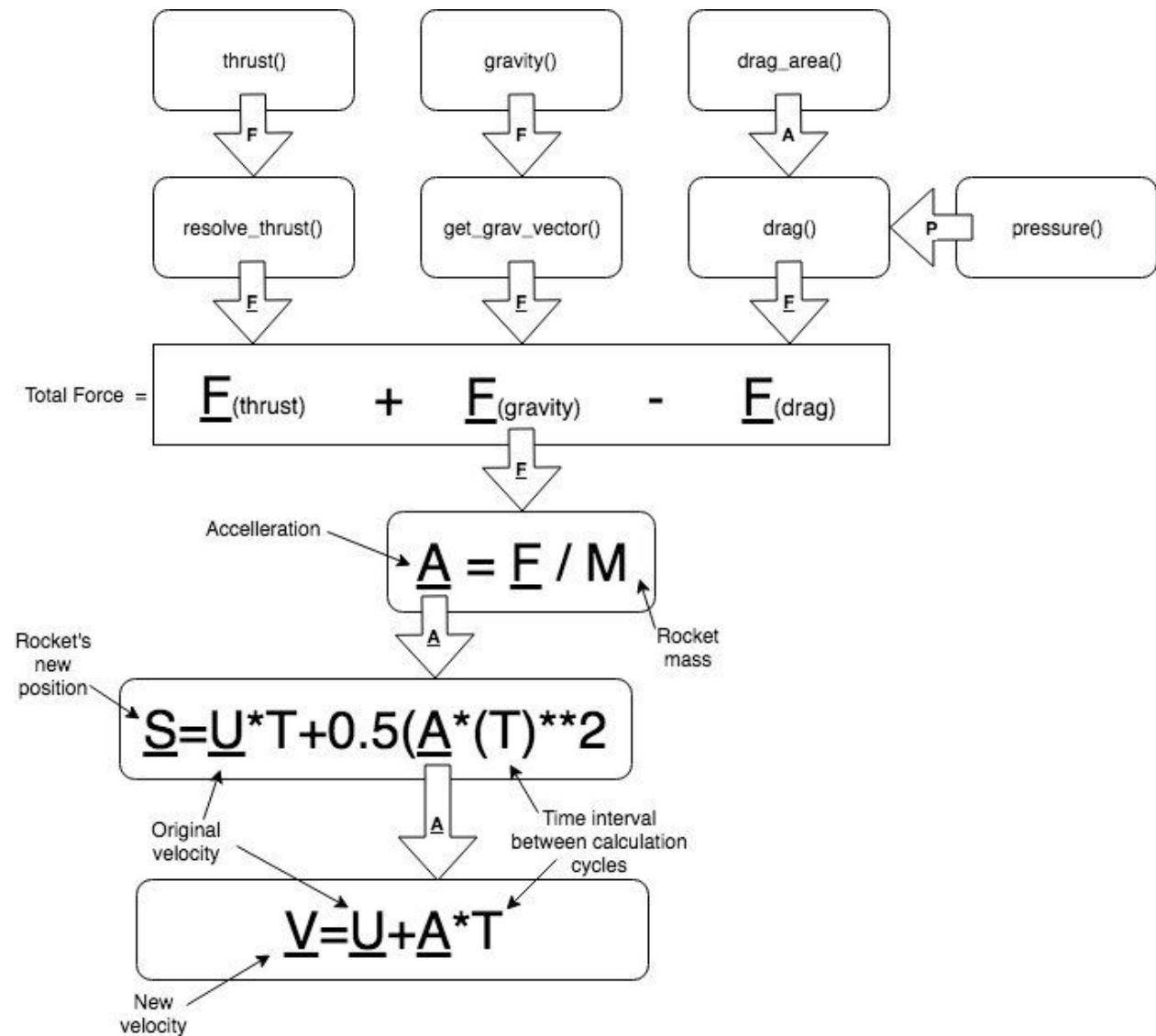
Other functions within the computational engine (resources.py)

Name	Parameters	Explanation
make_planet()	Planet, name, mass, radius, pos, Velocity, acceleration, pressure at sea level, Molar mass	<p>This function is used to instantiate a planet object dynamically so planets can be added in settings by the administrator either before, or during, the game. It means that objects within the game don't have to be hard coded.</p> <p>It also adds the planet (game object) to a list of all game objects to make it easier to keep track of them all.</p> <p><u>Pseudocode</u></p> <pre>Function make_planet(planet, name, m, r, p, V, a, p0, molMass): execute(str(name) + = Planet ('+str(mass) + ', '+str(radius) + ', '+ str(pos) + ', '+str(V) + ', '+str(a) + ', '+ 'p0' + ', '+str(molMass) + ')') execute('OBJECTS.append(['+str(name) +', "planet"])') This function makes use of an execute function, or exec() in python, which executes a string as if it were python code. So what would be run would simply be: Name = Planet(mass, radius, pos, V, a, p0, molMass) Where all the variables represent an actual number that would be entered</pre>
setup()		<p>This function sets up the game and is the main thing needed before the game runs as it instantiates and returns the Player object, as well as appending it to the list of current game objects.</p> <p><u>Pseudocode</u></p> <pre>OBJECTS = []</pre>

		<pre> Function setup(): make_planet(Earth...) OBJECTS.append([Earth, 'planet']) Player = Rocket(100, 3, 0.5...) OBJECTS.append([Player, 'player']) Return player ... indicates more parameters are entered than are shown. </pre>
run()	Planet object, Player object, time_interval	<p>This code is the actual code which is run every iteration to calculate the rocket's position, forces, trajectory etc. It utilizes all the mathematical functions seen previously. This function will be run by the server iteratively every time period t.</p> <p><u>Pseudocode</u></p> <pre> OBJECTS = [] data = [] Function run(planet, player, t): F = [0, 0, 0] player.F = [0, 0, 0] player.F = player.thrust(planet)*player.thm F = player.resolve_thrust() For i in OBJECTS: if i[1] == 'planet': F = [F[0]+player.get_grav_vector(i[0])[0], F[1]+player.get_grav_vector(i[0])[1], F[2]+player.get_grav_vector(i[0])[2]] if (i != 'player'): player.collision_test(i[0]) player.F = F player.a = [player.F[0]/player.mass, player.F[1]/player.mass, player.F[2]/player.mass] u = player.V player.V = [player.V[0]+player.a[0]*response_t, player.V[1]+player.a[1]*response_t, player.V[2]+player.a[2]*response_t] s = [(u[0]*response_t) + (0.5*player.a[0]*(response_t**2)), (u[1]*response_t) + </pre>

		<pre> (0.5*player.a[1]*(response_t**2)), (u[2]*response_t) + (0.5*player.a[2]*(response_t**2))] player.pos = [player.pos[0]+s[0], player.pos[1]+s[1], player.pos[2]+s[2]] data['p_acc'] = player.a data['p_vel'] = player.V data['p_pos'] = player.pos data['p_ang'] = player.angle data['p_fue'] = player.fuel data['p_thm'] = player.thm data['p_sta'] = player.status data['p_orb'] = player.orbit </pre> <p>As can be seen above, the majority of the pseudocode is calculations. At the very bottom the function sets some values of a dictionary to the player variables. This data is then sent to the client in the server script. The function does not return these values because</p>
update()	Player, data	<p>This function simply takes data from the client that have been changed, such as rocket angle and thrust modifiers, and adjust the variables in this script to match.</p> <p><u>Pseudocode</u></p> <pre> function update(player, data): global OBJECTS player.a = data['p_acc'] player.V = data['p_vel'] player.pos = data['p_pos'] player.angle = data['p_ang'] player.thm = data['p_thm'] </pre>

Mathematical Data Flow



2.7 Server Design

Variable list for app.py

Name	Type	Location	Description
CONFIG	Dictionary	Global	A dictionary of all the server settings
GAME	Dictionary	Global	A dictionary of all the game settings
JSON_AS_ASCII	Bool	CONFIG	Defines if JSON objects should be sent as Objects or Ascii strings
USE_X_SENDFILE	Bool	CONFIG	Defines if file-sending is enabled
SESSION_COOKIE_PATH	String	CONFIG	Defines where cookies are stored
SESSION_COOKIE_DOMAIN	String	CONFIG	Defines the domain of the session cookie
SESSION_COOKIE_NAME	String	CONFIG	Defines the session cookie name for the server
DEBUG	Boolean	CONFIG	Turns debug mode on/off
LOGGER_HANDLER_POLICY	String	CONFIG	Server setting
LOGGER_NAME	Boolean	CONFIG	Server log name
SESSION_COOKIE_SECURE	Boolean	CONFIG	Defines whether Flask should encrypt session cookies
SECRET_KEY	String	CONFIG	The encryption key for all data sent in and out of server

EXPLAIN_TEMPLATE_LOADING	Boolean	CONFIG	Server setting
MAX_CONTENT_LENGTH	Integer	CONFIG	The max length any string / object can be when sent from the server
PROPOGATE_EXCEPTIONS	String	CONFIG	Server setting
APPLICATION_ROOT	String	CONFIG	Sets where the server should run from in the local directory
SERVER_NAME	String	CONFIG	Changes the name of the server
PREFERRED_URL_SCHEME	String	CONFIG	Used to set up protocol-relative URLs
JSONIFY_PRETTYPRINT_REGULAR	Boolean	CONFIG	Server setting
TESTING	Bool	CONFIG	Server setting
PERMANENT_SESSION_LIFETIME	DateTime	CONFIG	Server setting
TEMPLATES_AUTO_RELOAD	String	CONFIG	Server setting
TRAP_BAD_REQUEST_ERRORS	Boolean	CONFIG	Server setting
JSON_SORT_KEYS	Boolean	CONFIG	Server setting
JSONIFY_MIMETYPE	String	CONFIG	Server setting
SESSION_COOKIE_HTTPONLY	Boolean	CONFIG	Server setting
SEND_FILE_MAX_AGE_DEFAULT	DateTime	CONFIG	Server setting
PRESERVE_CONTEXT_ON_EXCEPTION	String	CONFIG	Server setting
SESSION_REFRESH_EACH_	Boolean	CONFIG	Server setting

REQUEST			
TRAP_HTTP_EXCEPTIONS	Boolean	CONFIG	Server setting
GAME_NAME	String	GAME	The game name
DESCRIPTION	String	GAME	The game description
DIFFICULTY	Int	GAME	Game difficulty
ONLINE_MODE	Boolean	GAME	If the game should be online
WHITE_LIST	Boolean	GAME	If the game is white_list
BLACK_LIST	Boolean	GAME	If the game has a black_list
CHEATS	Boolean	GAME	If the game allows cheats
MAX_GAME_LENGTH	DateTime	GAME	The time before the game ends itself
BANNED_USERNAMES	List	GAME	Usernames that aren't acceptable
SCENARIO	Integer	GAME	A code to specify a scenario
ACTIVE	Boolean	GAME	If the simulation is active
USER_LIST	List	GAME	A list of users currently ingame
users	List		
meta	Dictionary		Meta data to be sent with each JSON file
start	datetime		The time at which the server started
player	Object		The player object
latest_pute	dictionary		The latest data fetched from

			resources.py
--	--	--	--------------

Flask App Routes

Upon receiving a request, my server will look at certain routes it can take depending on the request to act upon said request. E.g. If the server receives a GET request for /foo.html, it will find a html page loader route, execute the function within that route and return the result to the user in the form of a HTML page.

A lot of this is done by flask, however I have to define the routes myself. The routes are defined below:

Request: '/'

Methods:

Route to: index()

Description:

When the server receives a GET/POST request for the page at '/', it indicates that the client would like to receive the homepage of the website, so the function index() should render the homepage

Pseudocode:

```
function index():
    return render('index.html')
end function
```

Request: '/login'

Methods: POST

Route to: login()

Description:

When

Pseudocode:

```
function
```

Request: '/lobby'

Methods:

Route to: lobby()

Description:

When

Pseudocode:

```
function lobby():
```

Request: '/resources/img/<path:path>'

Methods:

Route to: img_route(path)

Description:

If an image is requested, it requires a different protocol to return the image to the client. For this reason any images that are stored on the server are confined to the directory '/resources/img' to make it clear to the server that it's requesting an image and needs an image protocol.

The image protocol requires the image to be encoded as a string of bytes in order to send it successfully, as it is treated as a file and not as a string, unlike a html document.

Pseudocode:

```
function img_route(path):  
    if path_exists(path) == True:  
        extension = split(path[-1].lower) // obtains the image extension  
        if extension == '.jpg' or '.png' or '.gif':  
            bytes = open(path, read)  
            return send_file(  
                encode_to_bytes(bytes.read()),  
                mimetype = 'image/' + string(extension)  
            )  
        end if  
    end if  
    else:  
        return "ERROR 404: "+string(path)+" doesn't exist  
    end else  
end function
```

Request: '/<path:path>'

Methods:

Route to: route(path)

Description:

If a page is requested that doesn't match a predefined route above, then the client is asking for something like a resource (E.g a JavaScript file). The server should load this path and return the result.

Pseudocode:

```
function route(path):  
    if path_exists(path) == True:  
        return render(path)  
    end if  
    else:  
        return "ERROR 404: "+string(path)+" doesn't exist  
    end else  
end function
```

SocketIO App Routes

Similarly, Socket.IO also uses routes to define what happens when it gets a request (when it receives data on its open websocket ports). The routes I use are only for data transfer, sending messages etc. It is logical to have different communication channels for different pages. For example you wouldn't want a list of who's currently connected sent with the same set of data that contains the calculations, since then the server would run the calculations every time a request for a connected user list is received, which isn't necessary and would slow down the program. The routes can be seen below:

Request: 'message'

Methods: web socket

Route to: handle_message (message)

Description:

This route is designed to take any message received on a non specific channel (namespace) and print it to screen, since it is non specific to any functions

Pseudocode:

```
function handle_message(message) :  
    print(message)
```

Request: 'message', namespace=/lobbu

Methods: web socket

Route to: handle_lobby_message (message)

Description:

This route is designed to take any message received on the lobby channel (namespace). To send to this channel the user will have to request '/lobbu' as the namespace.

Pseudocode:

```
function handle_lobby_message(message) :  
    send(LOBBY_DATA)
```

Request: 'message', namespace=/update

Methods: web socket

Route to: handle_incoming_data (message)

Description:

This route is designed to receive data (updated variables) from the client to update the calculation engine. This could be variables such as rocket angle, that have a direct impact on the calculations.

Pseudocode:

```
function handle_calc_message(message) :  
    resources.update(player, message)
```

Above, resources is the computational engine that contains the method update.

2.8 Client

Functions within the client

Name	Parameters	Explanation
(document).ready (anykey.js)	event	<p>This function is used in /index.html and is loaded from the javascript file anykey.js. It runs in the background and waits till it receives a key press. When this happens it will hide the 'press any key to continue' text and show a menu.</p> <p><u>Pseudocode</u></p> <pre>menu_flag = 0; \$(document).ready(function(evt) : if menu_flag == 0: if evt.keyCode == 8 OR 9 OR 10 OR .. #menu.show() #PAKTC_text.hide() menu_flag = 1)</pre>
join (anykey.js)		<p>This function is called when the join button is pressed in the menu in /index.html. It will hide the options menu and show the join_specific menu.</p> <p><u>Pseudocode</u></p> <pre>function join(): #menu.hide() #join-menu.show()</pre>
host (anykey.js)		<p>This function is called when the host button is pressed in the menu in /index.html. It will hide the options menu and show some text explaining how to run the server to host your own game</p> <p><u>Pseudocode</u></p> <pre>function join(): #menu.hide() #host-text.show()</pre>
about (anykey.js)		<p>This function is called when the about button is pressed in the menu in /index.html. It will hide the options menu</p>

		<p>and show some information about the game.</p> <p><u>Pseudocode</u></p> <pre>function join(): #menu.hide() #about-text.show()</pre>
setup (navball.js)		<p>This function uses three.js to setup the canvas for the nav-ball UI element. It defines the camera position and canvas size, and then adds the canvas onto the div with 'canvas' as its ID.</p> <p><u>Pseudocode</u></p> <pre>function setup(): scene = new THREE.scene() camera = new THREE.PerspectiveCamera() camera.position.z = 10; renderer = new THREE.WebGLRenderer() renderer.setSize(400, 400) canvas = document.getElementById('canvas' canvas.appendChild(renderer.domElement)</pre>
draw (navball.js)		<p>This function draws the 3D elements into the THREE.js scene. It defines what needs to be added to make the nav-ball</p> <p><u>Pseudocode</u></p> <pre>function draw(): geometry = new THREE.SphereGeometry() loader = new THREE.TextureLoader() loader.load('map.png', material = new THREE.MeshBasicMaterial({map: texture}) cube = new THREE.mesh(geometry, material)) scene.add(cube) pgeometry = new THREE.PlaneGeometry loader.load('shader.png', pmaterial = new THREE.MeshBasicMaterial({map: texture}) plane = new THREE.Mesh(pgeometry, pmaterial)) scene.add(plane)</pre>

		<pre>plane.position.set(0,0,0) plane.transparent = true</pre>
render (navball.js)		<p>This function renders the scene defined above.</p> <p><u>Pseudocode</u></p> <pre>function render(): renderer.render(scene, camera)</pre>
Render_navb all (navball.js)		<p>This function calls all the defined functions above. This function will be called when the user requests that the nav-ball be rendered in a modal</p> <p><u>Pseudocode</u></p> <pre>function render_navball(): setup() draw() render()</pre>
(document).re ady (lobby.html)		<p>This function is used to print the websocket information on /lobbu namespace to the screen. As usual when it connects it sends the server a message on the namespace to get the data flowing. Then when it receives a message on the namespace it prints it to screen. This function allows users to see what kind of game they're playing before they play it to plan their game. It also allows them to see who they're playing with. The function below also checks to see if the game is active, and if it is redirects the user to /ui.html</p> <p><u>Pseudocode</u></p> <pre>\$(document).ready(function(): lobby = io.connect('127.0.0.1:5000/lobbu') lobby.on('connect', function(): lobby.send('connected')) lobby.on('message', function(msg): if(msg.ACTIVE == true: window.redirect('/ui.html') user_table = '<table>' for i in msg.USERLIST.length: user_table = user_table + '<tr><td>' +msg.USERLIST[i]+ '</tr></td>'</pre>

		<pre> user_table = user_table + '</table>' document.getElementById('party').innerHTML = user_table lobby.send('message') Settings_table = '<table>' for i in msg.length -1: settings_table += <tr><td> settings_table+= i[enumerate(MSG)] settings_table+= '</td></tr>' document.getElementById('settings').innerHTML = settings_table </pre>
make (ui.html)	Id array, Value, decimal_point_location	<p>This function takes in an array of ids, value and decimal point of a numerical value meditor. It works by rounding the variable to 2 decimal places then splitting it into an array of characters in the value to be displayed on the NVE. It then performs checks to make sure that no value displays as null before writing these values to the corresponding id from the id array.</p> <p><u>Pseudocode</u></p> <pre> function make(id, value, dp): value = Math.round(value+100/100) value = string(value) value = value.split('') if value.length < id.length: value.unshift('0') if value[dp] == null: Value[dp] = '.' for i in id.length: if value[i] == null, value[i] == '0' document.getElementById(id[i]).innerHTML = value[i] </pre>
change (index.html)	type, power, direction	<p>This function changes the value of last_thm, a variable within the script by adding or subtracting a number to the power of 10. E.g. 1, 0.1, 0.01.</p>

		<p>It takes a type, power and direction. The type defines which if statement should be true which allows editing of different values other than last_thm. It also calls the make function defined above before sending the new data on a websocket with namespace /update to the server to update the variables inside the simulation.</p> <p><u>Pseudocode</u></p> <pre>Function change(type, power, direction): if type == 'thm': value = last_thm // defined earlier value = value+(1*power*direction) make(['thm1','thm2','thm3','thm4'], value, 2) last_thm = Math.round(value+100)/100 elseif ... update.send(lastest_data)</pre>
showModal (index.html)	id	<p>This function is called to show a specific modal when a button is pressed to make that modal appear in index.html. It also checks if the three.js navball scene (FDAl) has been called, and if so it calls the render function from navball.js</p> <p><u>Pseudocode</u></p> <pre>function showModal(id): win = document.getElementById(id) \$(win).css('z-index', ++zIndex) \$(win).show() If id=='fdai' AND fdai('rendered') == False: render_navball() fdai('rendered') == True</pre>
hideModal (index.html)	id	<p>This function is called to hide a specific modal when the X in the top right of that modal is clicked.</p> <p><u>Pseudocode</u></p> <pre>function hideModal(id): win = document.getElementById(id) \$(win).hide();</pre>

2.9 Development Test Plan

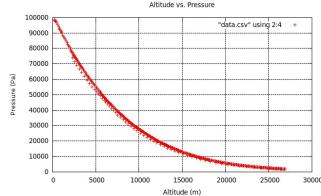
This is my plan for testing during the development of my software. Everytime a major change is made (github commit is made) all of the applicable tests from below will be tested against the new code.

E.g. I've altered server code so I'll test all the server tests, but not any graphical user interface tests. Everything will be tested throughout the course of my development

Description	Input	Expected output
Check that the UI page loads	Navigate to /ui.html once server is running	Page should load without error from a server. This is important to check after every change to make sure the page is unaffected. The JS error log should be empty
In /ui.html, check that modals appear	Load /ui.html	Modals should be visible on page
In /ui.html, check that modals are draggable	Click and drag the top bar of the modal	Modal should move with the cursor. This will improve usability by allowing a custom workspace
In /ui.html, check that modals are only draggable by the top bar	Click and drag top bar, and then click and drag main content area	Modal shouldn't move when dragged by content area. This is important to stop modals being accidentally dragged when interacting with them
In /ui.html, check that modals can overlap	Drag one modal on top of another modal and release mouse click	The modals should have no problem overlapping where one sits on top of another. This improves usability if there is a crowded workspace
In /ui.html, check that the currently dragged modal is in front	Drag one modal on top of another modal but do not release mouse click	The modal should appear on top of any modals behind so you can see what you're

		doing
In /ui.html, check that latest dragged modal is on top	Drag one modal on top of another and release mouse click	The latest dragged modal should have the highest z-index on the page. This allows the user to define which modals are most important if they've overlapping
In /ui.html, check modals become translucent when you dragged	Click and drag a modal without releasing mouse click	The modal should show a reduced opacity while dragged to show the user what is underneath whilst rearranging their workspace
In /ui.html, check modals cannot be dragged outside the visible page boundaries	Click and drag a modal past the page border	The modal edge should bump against the page border edge so no part of the modal is outside the page. This stops a modal being lost or unrecoverable
In /ui.html, check modals start hidden	Load the page	Modals shouldn't be visible to start. This encourages the user to use the navigation options later to set up their own custom workspace
In /ui.html, check that modals can be triggered by a javascript function	Create a button linking to a javascript show function for each modal and then click the button	The corresponding modal should appear when the corresponding modal is clicked. This is to confirm that the javascript show function will work at a later point.
In /ui.html, check that multiple modals can be open simultaneously	Click both buttons one after another	More than one modal should be visible at once to allow the user to be able to use multiple modals simultaneously
In /ui.html, check modals can be loaded in any order	Click the different modal buttons in different orders. Reload after each combination.	The modal should become visible in the order that the buttons were pressed. This is important for usability

In /ui.html, bootstrap styling should be applied across the entire page	Load page	Bootstrap styling should be obvious across every element of the page. This will make the user interface much more friendly
In /ui.html, check that a navigation bar can be added	Once navigation bar installed, load page	The navigation bar should appear at the top of the page as per design
Check that a flask server can be started	Launch the python app.py file from the terminal using python3	Look at the server terminal. The server should quote a debugger pin and a server address. Important to check the server works before continuing.
Check that the server can be accessed by correct IP and Port	When server is running, navigate to defined Flask server IP and port E.g (172.0.0.1:5555)	The address should load the defined index page without error. This is to make sure the server can be accessed remotely by the client
Check that the server receives a websocket message when the client page loads	Navigate to the server address	Look at server terminal. Printout 'User has connected' should appear. This is to ensure that the server can receive a trigger to start sending data
Check that the server sends the client messages in response to each received	Use an example like counting from 0-9 each time a message is received by the server and send this to the client. Load server page	Observe JS readout and check messages are received via websockets to make sure data can be communicated between client and server.
Check that the server sends and the client receives messages in sync	Use the 0-9 example from above	Watch both JS client console and server terminal to see if the messages (0-9) are in sync. This will be important for real time feedback in the game.
Check that client sends back message to server	In server, print out every time a message is received and repeat above test	Watch server terminal. Message should be received after every message sent. This makes sure the process

		is a continuous cycle
Check that the server can send longer messages	Set up server to send the accurate data and time using datetime function	Check the server terminal. Messages should still be sending as before. This verifies that strings can be send as well as integers
Check that the client can receive longer messages	Repeat above step	The JS client console should show messages being received as before. This verifies that strings can be received as well as integers
Check that the server can send objects over websockets	Create an object on the server and send it to the client.	Check server terminal. Objects should be sent. This is important as it means multiple bits of data can be sent in one packet.
Check that the client can receive objects	Repeat above step	Check JS client console. Objects should be printed and obvious. This is important because it means the client can receive multiple pieces of information at once
Check that the client can parse JSON Objects	Get the client to print the object elements separately	Each element of the Object should be printed clearly and separately to prove that the client can parse JSON objects.
Check that the air pressure calculation functions correctly	Plot a graph of height against air pressure using matplotlib	<p>The air pressure calculation should show a logarithmic curve like so:</p>  <p>This proves that the function works</p>
Check that the thrust calculation functions correctly	Use values from the following problem:	It should return a value of 96500 for F to prove that the

	<p>PROBLEM 1.1</p> <p>A spacecraft's engine ejects mass at a rate of 30 kg/s with an exhaust velocity of 3,100 m/s. The pressure at the nozzle exit is 5 kPa and the exit area is 0.7 m². What is the thrust of the engine in a vacuum?</p> <p>SOLUTION,</p> <p>Given: $q = 30 \text{ kg/s}$ $V_e = 3,100 \text{ m/s}$ $A_e = 0.7 \text{ m}^2$ $P_e = 5 \text{ kPa} = 5,000 \text{ N/m}^2$ $P_a = 0$</p> <p>Equation (1.6),</p> $F = q \times V_e + (P_e - P_a) \times A_e$ $F = 30 \times 3,100 + (5,000 - 0) \times 0.7$ $F = 94,500 \text{ N}$	thrust function works
Check that the atmospheric drag calculation function correctly	Run the equation using values from the above test	The atmospheric drag equation should return a sensible positive result that is lower than the value of F from above test. This proves that the atmos drag function works
Check that the gravitational attraction calculations work	Run the equation using the mass of the earth and 10kg with a radius of 1E30	The equation should return a force of $\sim 4E-45$. This proves that the gravitational attraction function works
Check that all major mathematical functions integrate together	Plot the returned values of functions on a graph	The function when plotted should show a steep drop and then a logarithmic gradual decline. This is to check that all the functions work together
Check that all the equations work together to produce a force value	Run the equations to find F	The functions should integrate correctly and create a 3D vector force in the positive Y direction to make sure that Force (the main component) can be calculated as designed.
Check a 3D position system can be used and functions correctly with the vector equations	Create a position system then plot a graph of every data point on an XYZ plane	The position system should record where the rocket is at every iteration. It should clearly show the rocket moving in the +Y direction due to the +Y force. This must be checked to make sure that a position system can be used with the mathematical vector functions
Check that using the position	Run the calculations with	When plotted it should show

system a gravitational orbit can be achieved using the gravitational equation	values that would force an orbit and plot a graph of every data point on an XYZ plane	a clear elliptical shape existing in only 2 planes. This is to verify that the position system works with the gravitational equations
Check that three.js pages can be loaded on a Flask server	Load a page with a three.js canvas running on the server	The page should load without error to check that three.js (and thus WebGL) works on the server
Check that three.js Nav-ball loads	Create a three.js canvas and nav-ball and load page	The three.js canvas and navball elements should be visible on the page
Check that three.js Nav-ball can be rotated using javascript	Using the JS console, set the navball X,Y,Z rotation to different values	The navball should rotate
Check that the sphere map can be updated	Change the navball sphere map from old to new and load page	The navball should show an updated surface map to check that maps are loaded from the correct directory and that the new map can be used.
In /ui.html, check a nav-ball button can be added to navigation bar	Add a nav-ball button and load ui.html page	A nav-ball button should appear in a drop down menu of the navbar to make sure the UI functions with new changes
In /ui.html, check a three.js nav-ball canvas can be integrated into a modal	Link nav-ball button to a Modal containing three.js navball	The three.js navball and canvas should appear within the modal and can be dragged with the modal. This is an important usability feature.
In /ui.html, check that nav-ball can rotate according to defined server data over websockets	Send increasing data over websockets continuously and apply this data to the nav-ball rotation using the javascript function	The navball should spin in accordance to websocket data. This ensures that websocket data can be used to influence and change three.js entities.
Check that the server can	Change settings in	Check the server terminal

parse text and load settings from CONFIG.txt into a dictionary	CONFIG.txt and print when server starts.	readout. The printed settings should match those identically in the CONFIG.txt file. This test makes sure that settings can be loaded correctly.
Check that an index page loads	Navigate to {IP}:{PORT} /	The page should load without error to make sure that rendering index.html causes no server issues.
Check that the index page renders the defined index page	Navigate to {IP}:{PORT} /	The correct index.html page should be rendered at / to make sure the flask app routes are working correctly.
Check that the index page has a functioning press any key to continue feature	Press every standard keyboard key, reloading each time	Each key should trigger the Join/Host/About menu to appear. It's important to check everykey to make sure the function works across the board.
Check join button functions	Click join button	A text box should take its place. The join button is needed to function to select what mode you want.
Check name text-field works	Enter different Unicode & Ascii symbols in the text field	Every character should work fine. This is to make sure the game works in different countries with different keyboards.
Check the final join button redirects to lobby	Click join button in /	The page should redirect to /lobby. This test is needed to verify that the page routing works correctly
Check /lobby loads correctly	Click join button in /	The page should load with zero errors to check that the page redirect is correct and accepted by the server
Check /lobby renders lobby page correctly	Click join button in /	The lobby.html page should be rendered at /lobby to make sure page routing

		works for the lobby
Check lobby displays game settings	Load /lobby	The lobby should show the current game settings to help the players understand how their specific game will work so they can develop a strategy going in
Check lobby displays currently connected users	Load /lobby	The lobby should show a table of all the currently connected users, to let the players know when to start the game and who they're playing with
Check that the server recognises a user has connected	Join /lobby from /	Observe server terminal readout and the name of the connected user should be printed when they connect. This checks that the /lobby namespace is working and messages can be transmitted on a separate websocket channel
Check the server allows multiple users to be added	Join /lobby from / in different tabs simultaneously	Each username should be printed in the server terminal in the order they join. This makes sure that multiple users can join
All clients update data instantly	Join /lobby in a new tab	Observe both old and new tabs and both username lists should update simultaneously in real time.
Check that orbit checking function works	Set variables to be correct for a newtonian orbit and run function. Then set variables to be a non orbit and run function. Print the data points on an XYZ plane to visualise rocket path	Variables for newtonian orbit should make the orbit check function return true. The XYZ graph should confirm this. Variables for non-orbit should make the orbit check function return false. The XYZ should confirm this. This check is important to make sure that the only current game

		objective works as needed.
Check resources.py and all its functions work together as expected	Run resources.py printing key variables and plotting XYZ data points	Resources.py should run with no errors and the produced key variables should match with the XYZ position graph
Check numerical variable editors display correctly	Load a numerical variable editor	The NVE should display correctly with arrow buttons aligned correctly over each digit. It is important to check this looks right before continuing so it won't be confusing for users.
Check numerical variable editors can change a specific digit by a power of 10	Click different digit's up and down buttons	The corresponding digit should change up or down by 1 to make sure that the NVE works as it should.
Check JS records correct variable to match numerical editor	Print value in JS console everytime change function is called	The number printed in the JS console should match the number on screen precisely to make sure it can be sent to the server correctly
Check numerical variable editor can handle 0 values	Change the value to 26.0	The digit that is 0 should show the 0 character and not Null, Non, NaN, or Undefined to make sure it displays correctly.
Check numerical variable editor can't go outside it's range	Enter 106.0 (> 100)	The NVE shouldn't overflow outside its boundaries. The decimal point shouldn't move, instead the value shouldn't be allowed to be increased over 100 to make sure it displays correctly.
Check numerical variable editor can handle change in decimal point position	Enter 6.09	The NVE should not move the decimal point to compensate but instead a 0 should be added to the start of the value to make it easier to read and avoid user confusion.

Check numeric variable editor can handle negative values	Enter -3.9	The NVE should add a negative sign at the beginning instead of changing one of the digits to a negative sign to avoid confusion for the end user.
Check that numerical variable editors for angle changes server data	Using update function in app.py, change rocket angle in the UI	Observing the server readout in /ui.html should show that the value (pulled straight from server over websockets) changes in accordance with the NVE value. This is to check that the data can be sent from client and directly edit angle variables in the computational engine (resources.py)
Check that numerical variable editor for thrust multiplier changes server data	Using update function in app.py, change thrust modifier in the UI and observe the server readout in /ui.html	Observing the server readout in /ui.html should show that the value (pulled straight from server over websockets) changes in accordance with the NVE value. This is to check that the data can be sent from client and directly edit thrust modifier variables in the computational engine
Check the server receives new data from client	Alter client variables	The server terminal should print 'data received' with a JSON object of all the data including the newly updated data to show that new data can be received on the update namespace websocket channel.
Check that using multithreading has no effect on the server starting	Start the server	The server should start correctly with an IP and debugger PIN to show that it still works regardless of the multithreading setup
Check that multithreading has no effect on the data the	Start server. Load /ui.html	The server readout in /ui.html should be full and every value

client receives		should have a value instead of an empty string or error to show that data can still be sent between the two threads (server and computational)
Check that multithreading has no effect on the client's ability to change server data	Load /ui.html and change data in client ui.	The server readout in /ui.html should still reflect the value on the NVE to make sure data can pass between threads fine in the server without any adverse effects.

2.10 Final Testing Plan

Test No.	Description	Input	Expected output
1	Admin menu	Run run.py with python3	Admin menu should open without error. This is to check that there are no errors in the python code at the end of development.
2	Admin title	Run run.py with python3	Text 'M.C.S 3000' should appear in ascii art. This proves that the code actually runs and gives the admin user a nice introduction to the program.
3	Admin help	'help()' in run.py	A help menu should appear with a list of different functions you can type into the terminal to perform different functions. It's important to check this to make sure the user can get the help they need to operate the program.
4	Server start	'start()' in run.py	The terminal should print the CONFIG.txt options and end with a debugger pin and the server address. This test

			shows that the subprocess has started correctly and that the server can still boot correctly in this new version of the code. If you check the LAN in wireshark you should see a server on the server IP and port.
5	Server stop command	'stop()' in run.py	The terminal should verify that you want to kill the python process. This check is required to make sure the user has an option to abort the stop procedure since it will be killing all python services.
6	Server stop	'stop()' in run.py then 'Y'	The terminal window should close along with the server. Use wireshark to verify that no server is running on the network. This check ensures that the server can be shut down and that the sub-process can be killed using the stop() command.
7	Server abort stop	'stop()' in run.py then 'N'	The terminal window should not close and the server should continue running as normal. This step is important to make sure the stop() function options work correctly and don't kill all python processes unnecessary..
8	Server invalid input stop	'stop()' in run.py then 'n', 'y', 'yes', 'no' and 'k'	The terminal should ask you to type Y or N again. This is important to give the user feedback and not to take any input. The check is to make sure that it doesn't continue with any other processes.
9	Server IP Change	Change IP in CONFIG.txt then start server	The start server text should display the new IP instead of 127.0.0.1, and if you navigate to

			<p>the new IP the server should be running there.</p> <p>You can verify this with wireshark. This test is important because some users don't want to run on their local machine and may wish to open the game up to WAN not LAN so the IP change feature must be tested.</p>
10	Server Port Change	Change Port in CONFIG.txt then start server	<p>The start server text should display the new port instead of 127.0.0.1, and if you navigate to the new IP the server should be running there.</p> <p>You can verify this with wireshark. This test is important because some users don't want to run on their local machine and may wish to open the game up to WAN not LAN so the port change feature must be tested.</p>
11	Server IP Change on start	Run server with 'start('192.168.10.142')	<p>The start server text should display 192.168.10.142 instead of 127.0.0.1, and if you navigate to the new IP the server should be running there.</p> <p>You can verify this with wireshark. This test is important because some users don't want to run on their local machine and may wish to open the game up to WAN not LAN so the IP change feature must be tested.</p>
12	Server Port Change on start	Run server with 'start('127.0.0.1', '8888')	<p>The start server text should display the new port 888 instead of 5000, and if you navigate to the new IP the server should be running there.</p> <p>You can verify this with</p>

			wireshark. This test is important because some users don't want to run on their local machine and may wish to open the game up to WAN not LAN so the port change feature must be tested.
13	User connect to server index	Navigate to {ip}:{port}/	While the server is running the index page should load and the server terminal should display a GET request. This check makes sure that the server is accessible at the correct address via browser.
14	Check that an index page loads	Navigate to {ip}:{port}/	The index page should be served by the server and the browser should show the page loading with no critical errors in the JS console or log. This check is needed to make sure there are no page errors in the final version
15	Check that the index page renders the defined index page	Navigate to {ip}:{port}/	The index page served should be index.html. This makes sure that Flask is rendering the correct page and the app route is correct.
16	Check that the index page has a functioning press any key to continue feature	On /index.html, press every standard key on a British standard keyboard. Refresh page each time.	Each key should trigger the join/host/about menu to appear. This makes sure that there isn't a key that you can press that won't trigger the menu and that the ANY part of press any key holds true.
17	Check join button functions	Click the Join button on /	The menu should disappear and show a text-field. This is to check that the javascript functions are called correctly and that this button functions as needed in the final product.
18	Check host button	Click the Host button	The page should display an

	functions	on /	alert saying you will be redirected to a different page, and then the project github page should load. This makes sure that the host button works as needed and the page redirects successfully.
19	Check about button functions	Click the About button on /	The page should display an alert giving some about text. This test is needed to make sure the about button works so confused users can get more information.
20	Check name text-field works	Type all standard western keyboard keys into the text box	The characters should appear in the text box as you type them. This makes sure that the textbox can display ASCII correctly.
21	Check name text-field can't be empty	Click join with the text field empty	Nothing should happen. This makes sure the javascript check works and the user can't submit a blank name to the server. I need to test this to ensure this key validity check works.
22	Check the final join button redirects to lobby	Click the join button	The page should be directed to /lobby. This check is needed to make sure the page still directs in the final version.
23	Check /lobby loads correctly	Click join button in /	The page should load with zero errors to check that the page redirect is correct and accepted by the server
24	Check /lobby renders lobby page correctly	Click join button in /	The lobby.html page should be rendered at /lobby to make sure page routing works for the lobby
25	Check lobby displays game settings	Load /lobby	The lobby should show the current game settings to help

			the players understand how their specific game will work so they can develop a strategy going in
26	Check lobby displays currently connected users	Load /lobby	The lobby should show a table of all the currently connected users, to let the players know when to start the game and who they're playing with
27	Check that the server recognises a user has connected	Join /lobby from /	Observe server terminal readout and the name of the connected user should be printed when they connect. This checks that the /lobby namespace is working and messages can be transmitted on a separate websocket channel
28	Check the server allows multiple users to be added	Join /lobby from / in different tabs simultaneously	Each username should be printed in the server terminal in the order they join. This makes sure that multiple users can join
29	All clients update data instantly	Join /lobby in a new tab	Observe both old and new tabs and both username lists should update simultaneously in real time.
30	Check client redirects to /ui.html when game starts	Keep client on /lobby and start game using run.py 'start_game()'	The window should automatically redirect to /ui.html. This test needs to be completed to make sure that the game can be started remotely from the server in the final version.
31	Check main websocket channel is running correctly	Navigate to /ui.html with server running	In the top right the current game time should be continuously running. This means the client and server are communicating over websockets correctly. This

			check is one of the most important because the entire game relies on the use of this websocket channel so it is very important to check that it works in the final version.
32	Check /lobbu websocket channel is running correctly	Navigate to /lobby via /index.html with server running	When the lobby has been joined the username entered in /index.html should appear in the user's list. This string can only be pulled from the server so it indicates that the name has been sent to the server and the server has responded both on the correct /lobbu channel. This check is necessary as it means the namespace routes are set up correctly
33	Check /update websocket channel is running correctly	Change rocket angle in /index.html with server running	When the rocket angle has been changed check the server terminal readout. This should print the data received. The data received should match the new rocket angle and thus proves that the /update channel is working fine. This channel is important because it's the only way the user can interact with the game simulation. Without it there wouldn't be a game.
34	In /ui.html, check modals start hidden	Load the page	Modals shouldn't be visible to start. This encourages the user to use the navigation options later to set up their own custom workspace
35	In /ui.html, check modals cannot be dragged outside the visible page boundaries	Click and drag a modal past the page border	The modal edge should bump against the page border edge so no part of the modal is outside the page. This stops a modal being lost or unrecoverable

36	In /ui.html, check modals become translucent when you dragged	Click and drag a modal without releasing mouse click	The modal should show a reduced opacity while dragged to show the user what is underneath whilst rearranging their workspace
37	In /ui.html, check that modals are draggable	Click and drag the top bar of the modal	Modal should move with the cursor. This will improve usability by allowing a custom workspace
38	In /ui.html, check that modals are only draggable by the top bar	Click and drag top bar, and then click and drag main content area	Modal shouldn't move when dragged by content area. This is important to stop modals being accidentally dragged when interacting with them
39	In /ui.html, check that modals can overlap	Drag one modal on top of another modal and release mouse click	The modals should have no problem overlapping where one sits on top of another. This improves usability if there is a crowded workspace
40	In /ui.html, check that the currently dragged modal is in front	Drag one modal on top of another modal but do not release mouse click	The modal should appear on top of any modals behind so you can see what you're doing
41	In /ui.html, check that latest dragged modal is on top	Drag one modal on top of another and release mouse click	The latest dragged modal should have the highest z-index on the page. This allows the user to define which modals are most important if they've overlapping
42	In /ui.html, check that multiple modals can be open simultaneously	Click both buttons one after another	More than one modal should be visible at once to allow the user to be able to use multiple modals simultaneously
43	In /ui.html, check modals can be loaded in any order	Click the different modal buttons in different orders. Reload after each combination.	The modal should become visible in the order that the buttons were pressed. This is important for usability
44	In /ui.html, check that	Load the page	The navigation bar should be

	the navigation bar loads		visible at the top of the page. This makes sure the html and css work correctly.
45	In /ui.html, check that the navigation bar has a functional drop down menu	Click one of the navigation bar headings	A drop down menu should appear from that specific heading to allow for a more categorical range of options to create a more organised system
46	In /ui.html, check that the navigation bar buttons can trigger a modal	Click one of the buttons in the navigation bar drop down menu	The button should trigger the corresponding modal to become visible so it can be used
47	In /ui.html, check that an exit button appears in the modal handle	Open a modal using the nav bar	An exit [x] button should appear in the top right of the handle. This is an important UI feature to make the program easier to use.
48	In /ui.html, check that the exit button hides the modal	Click the [x] exit button	The Modal should disappear (be hidden) to allow the user to clean up / reorganise their work space
49	In /ui.html, check that nav-ball can rotate according to defined server data over websockets	Send increasing data over websockets continuously and apply this data to the nav-ball rotation using the javascript function	The navball should spin in accordance to websocket data. This ensures that websocket data can be used to influence and change three.js entities.
50	Check numeric variable editor can handle negative values	Enter -3.9	The NVE should add a negative sign at the beginning instead of changing one of the digits to a negative sign to avoid confusion for the end user.
51	Check that numerical variable editors for angle changes server data	Using update function in app.py, change rocket angle in the UI	Observing the server readout in /ui.html should show that the value (pulled straight from server over websockets) changes in accordance with the NVE value. This is to check

			that the data can be sent from client and directly edit angle variables in the computational engine (resources.py)
52	Check that numerical variable editor for thrust multiplier changes server data	Using update function in app.py, change thrust modifier in the UI and observe the server readout in /ui.html	Observing the server readout in /ui.html should show that the value (pulled straight from server over websockets) changes in accordance with the NVE value. This is to check that the data can be sent from client and directly edit thrust modifier variables in the computational engine
53	Check the server receives new data from client	Alter client variables	The server terminal should print 'data received' with a JSON object of all the data including the newly updated data to show that new data can be received on the update namespace websocket channel.

Development Log

Github

Throughout the development of my project I have used a version control service called Github. I did this because it makes the development of my software a lot easier. Everytime I finish a major change I 'commit' the code to github along with screenshots and a description. When I look back on the code I can see changes and roll back the code to a previous version at any point if something goes wrong, or just for testing purposes.

Github: Branches

My development is split into two 'branches'. The first branch is my **Master Branch**. This consists of all the code for my project that I intend to be in the final product.

My second branch is the **Experiments Branch**. This is where i build my prototypes and experiments that I plan to use later and merge into the master branch.

Github: Issues

I also have **Issues**. Anything marked as an Issue shows a large problem I had with my code that I had to work through, and it shows my progress and how I work through problems to find a viable solution.

Testing

After every large commit you can find a test that took place during the development of this chunk of my project. For my testing plan refer to the Design section.

Started development of User Interface

[Browse files](#)

Added draggable windows using JQuery UI.

Used JQuery UI because there is nothing extra to install (using a CDN), hence is compatible with more systems. It's also very versatile and easy to work with, as well as supporting mobile-friendly web pages.

by master

 meowterspace committed on 26 Oct 2017

1 parent 37f2f84 commit f501acc267df877a2a6f8098b4e50a358a88c43d

 Showing 1 changed file with 72 additions and 0 deletions.

[Unified](#) [Split](#)

72  ui.html

```
...  ... @@ -0,0 +1,72 @@
1  +<html>
2  +  <head>
3  +
4  +  <title>Main</title>
5  +
6  +  <style type="text/css">
7  +    body { background-color: #c4c4c4;
8  +           margin: 0px;
9  +           height: 100%;
10 +          }
11 +  .window { border: 1px solid #000000;
12 +           background-color: #ffffff;
13 +          }
14 +  .handle { background-color: #00ff00;
15 +           margin-top: 0em;
16 +          }
17 +  #win1 { width: 200px;
18 +           height: 400px;
19 +           position: relative;
20 +           left: 5px; /* START POSITION ▲ ▼ */
21 +           top: 5px;
22 +           clear: both;
23 +           z-index: 1;
24 +          }
25 +  #win2 { width: 400px;
26 +           height: 200px;
27 +           position: relative;
28 +           left: 200px; /* START POSITION ▲ ▼ */
29 +           top: -400px;
30 +           clear: both;
31 +           z-index: 1;
32 +          }
33 +
34 +  </style>
35 +
36 +  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
37 +  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
38 +  <script>
39 +
40 /* This makes #winx draggable only by handle(class .handle)
41 + with a change in opacity on event and ensures most recently dragged
42 + has the highest z-index */
43 + $(function() {
44 +   var zIndex = 1;
45 +
46 +   $("div[id*='win']").draggable({
47 +     handle: $(".handle"),
48 +     opacity: 0.8,
49 +     start: function(event, ui) {
50 +       $(this).css("z-index", ++zIndex);
51 +     }
52 +   });
53 + }
54 + 
```

```
52          }
53      });
54  });
55  </script>
56
57
58  </head>
59
60  <body>
61
62  <div id="win1" class="window">
63      <p class="handle">Window 1</p>
64      <div class="content">CONTENT</div>
65  </div>
66  <div id="win2" class="window">
67      <p class="handle">Window 2</p>
68      <div class="content">CONTENT</div>
69  </div>
70
71
72  </body>
```

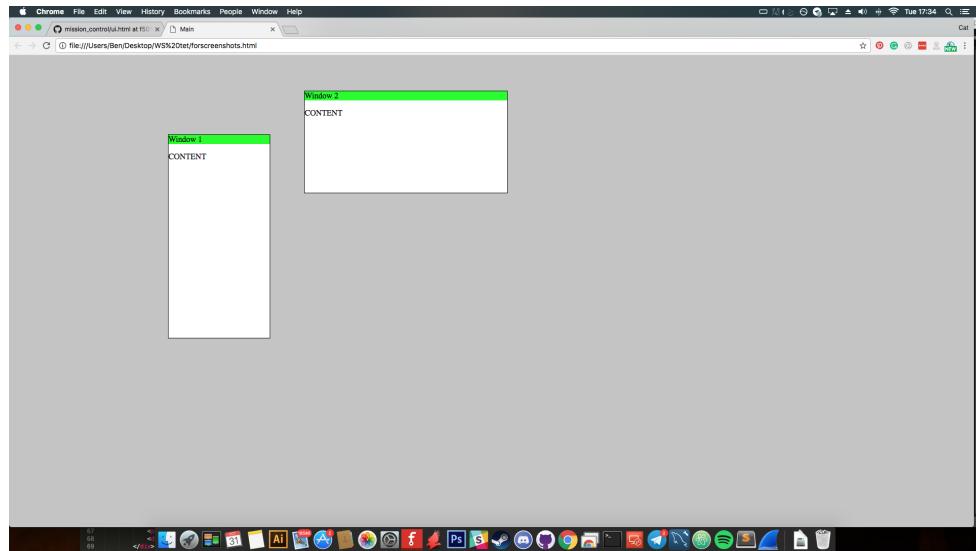
1 comment on commit [f501acc](#)



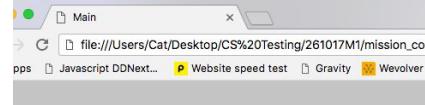
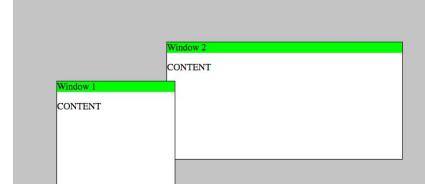
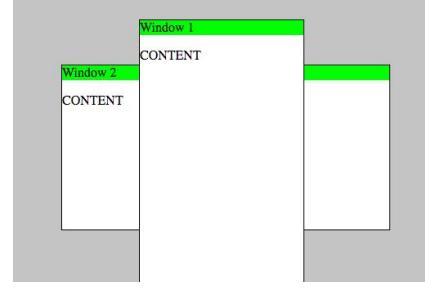
meowterspace commented on f501acc on 31 Oct 2017

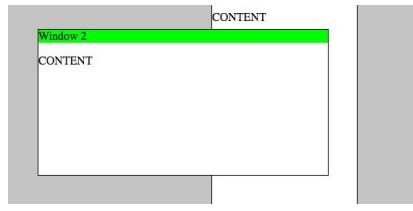
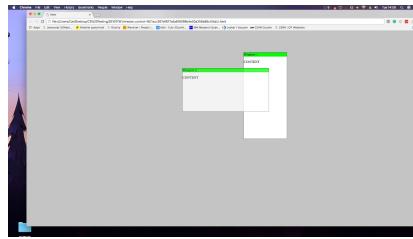
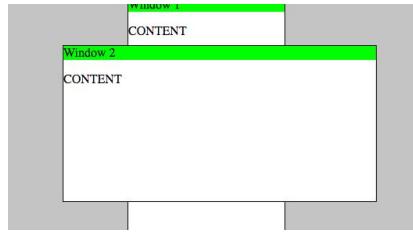
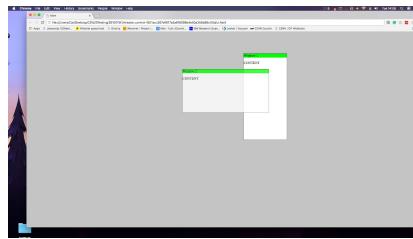
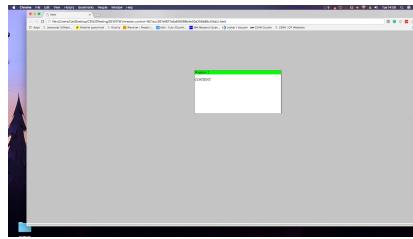
Owner

Preview



261017M1

Test	Pass/Fail	Description	Evidence
Check that the UI page loads	PASS	Page should load without error from a server. This is important to check after every change to make sure the page is unaffected. The JS error log should be empty	
In /ui.html, check that modals appear	PASS	Modals should be visible on page	
In /ui.html, check that modals are draggable	PASS	Modal should move with the cursor. This will improve usability by allowing a custom workspace	
In /ui.html, check that modals are only draggable by the top bar	PASS	Modal shouldn't move when dragged by content area. This is important to stop modals being accidentally dragged when interacting with them	

In /ui.html, check that modals can overlap	PASS	The modals should have no problem overlapping where one sits on top of another. This improves usability if there is a crowded workspace	
In /ui.html, check that the currently dragged modal is in front	PASS	The modal should appear on top of any modals behind so you can see what you're doing	
In /ui.html, check that latest dragged modal is on top	PASS	The latest dragged modal should have the highest z-index on the page. This allows the user to define which modals are most important if they've overlapping	
In /ui.html, check modals become translucent when you dragged	PASS	The modal should show a reduced opacity while dragged to show the user what is underneath whilst rearranging their workspace	
In /ui.html, check modals cannot be dragged outside the visible page boundaries	FAIL	The modal edge should bump against the page border edge so no part of the modal is outside the page. This stops a modal being lost or unrecoverable	

Browse files

Converted Windows to Modals

I've added a show functionality to the windows so they can start hidden and appear only when requested.

This functionality will be used later when the user can select only the important windows to view, to create the best layout to suit their personal needs

` master

meowterspace committed on 26 Oct 2017

1 parent f501acc commit d00190a33c0fdd3e543a591dcf5aa164ab043fd8

Showing 1 changed file with 26 additions and 9 deletions.

Unified Split

35 ████ ui.html

```

3 3
4 4
5 5
6 + <title>Main</title>
7 + <!-- INCLUDE THE JQUERY LIBRARIES VIA CDN -->
8 + <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
9 +
10 + <style type="text/css">
11 -         body { background-color: #c4c4c4;
12 -             body { background-color: #c4c4c4;
13 -                 margin: 0px;
14 -                 height: 100%;
15 -             }
16 -         }
17 +         .window { border: 1px solid #000000;
18 +             background-color: #ffffff;
19 +             .handle { background-color: #00ff00;
20 +                 margin-top: 0em;
21 -             #win1 { width: 200px;
22 -                 #win1 { width: 200px;
23 -                     height: 400px;
24 -                     position: relative;
25 -                     left: 5px; /* START POSITION ▲ ▼ */
26 -                     top: 5px;
27 -                     clear: both;
28 -                     z-index: 1;
29 -                     display: none;
30 +             #win2 { width: 400px;
31 +                 #win2 { width: 400px;
32 -                     height: 200px;
33 -                     position: relative;
34 -                     left: 200px; /* START POSITION ▲ ▼ */
35 -                     top: -400px;
36 -                     clear: both;
37 +             }
38 +             }
39 +
40         </style>
41
42 -         <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
43 -         <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
44 -         <script>
45 -             /* This makes #winx draggable only by handle(class .handle)
46 -             with a change in opacity on event and ensures most recently dragged
47 -             has the highest z-index */

```

```

45  +             /* This makes #winx draggable only by handle(class .handle)
46  +             with a change in opacity on event and ensures most recently dragged
47  +             has the highest z-index */
48  +             $(function() {
49  +                 var zIndex = 1;
50
51  +                 });
52
53  +             </script>
54
55  +             // Function to make the window visable on event
56
57  +             function showModal(id) {
58  +                 console.log(id);
59  +                 var win = document.getElementById(id);
60  +                 $(win).show();
61  +             };
62
63  +             </script>
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86  +             <button onclick="showModal('win1');">Window 1</button>
87  +             <button onclick="showModal('win2');">Window 2</button>
88
89

```

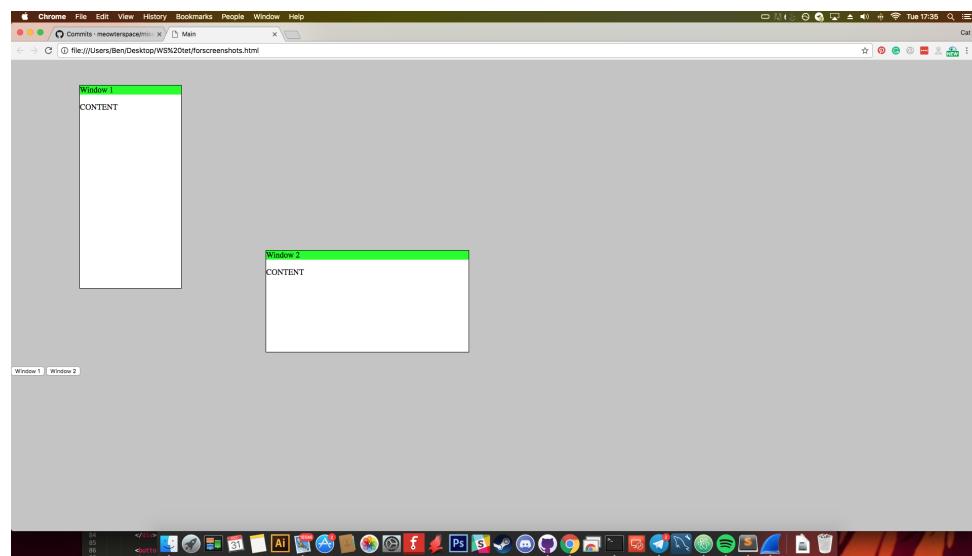
1 comment on commit [d00190a](#)



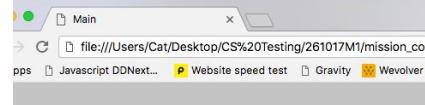
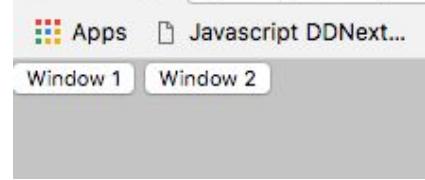
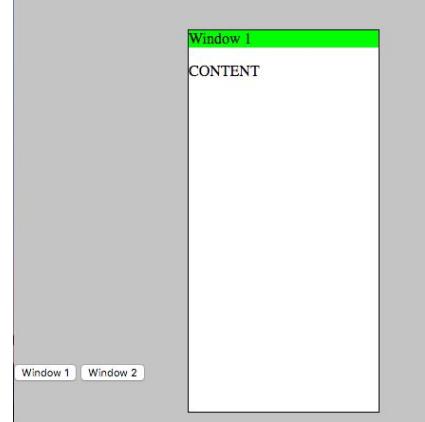
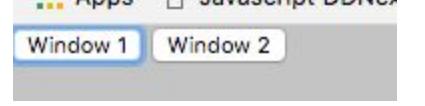
meowterspace commented on [d00190a](#) on 31 Oct 2017

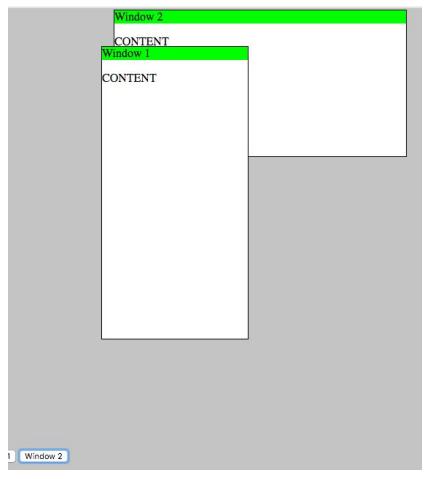
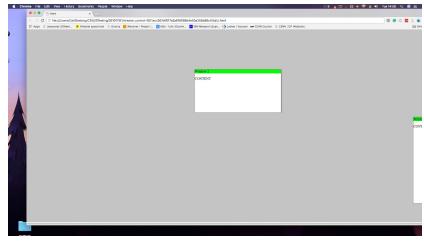
Owner

Preview



261017M2

Test	Pass/Fail	Description	Evidence
Check that the UI page loads	PASS	The page should load when the file opens without error.	
In /ui.html, check modals start hidden	PASS	Modals shouldn't be visible to start. This encourages the user to use the navigation options later to set up their own custom workspace	
In /ui.html, check that modals can be triggered by a javascript function	PASS	<p>The corresponding modal should appear when the corresponding modal is clicked. This is to confirm that the javascript show function will work at a later point.</p> <p>Here the javascript function is triggered by a button ></p>	 

In /ui.html, check that multiple modals can be open simultaneously	PASS	More than one modal should be visible at once to allow the user to be able to use multiple modals simultaneously	
In /ui.html, check modals can be loaded in any order	FAIL	<p>The modal should become visible in the order that the buttons were pressed. This is important for usability</p> <p>-----</p> <p>Modal number 2 can only appear after modal 1.</p>	
In /ui.html, check modals cannot be dragged outside the visible page boundaries	FAIL	The modal edge should bump against the page border edge so no part of the modal is outside the page. This stops a modal being lost or unrecoverable	

Started to use Bootstrap for styling

[Browse files](#)

I've included the Bootstrap CSS CDN which should make the user interface look a lot nicer and save a lot of time when making more complex features.
This is important for ease of use as well as user enjoyability.

by master

 meowterspace committed on 26 Oct 2017

1 parent d00190a commit f18e25dc43a442db0b3e37262842d0bec2b97d94

 Showing 1 changed file with 13 additions and 10 deletions.

[Unified](#) [Split](#)

23  ui.html

```
7 7          <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
8 8          <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
9 9
10 10         +
11 11         +
12 12         +
13 13         <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
14 14         <style type="text/css">
15 15             body { background-color: #c4c4c4; margin: 0px; }
16 16
17 17
18 18
19 19
20 20
21 21
22 22
23 23
24 24
25 25
26 26
27 27
28 28
29 29
30 30
31 31
32 32
33 33
34 34
35 35
36 36
37 37
38 38
39 39
40 40
41 41
42 42
43 43
44 44
45 45
46 46
47 47
48 48
49 49
50 50
51 51
52 52
53 53
54 54
55 55
56 56
57 57
58 58
59 59
60 60
61 61
62 62
63 63
64 64
65 65
66 66
67 67
68 68
69 69
70 70
71 71
72 72
73 73
74 74
75 75
76 76
77 77
78 78
79 79
80 80
81 81
82 82
83 83
84 84
85 85
86 86
87 87
88 88
89 89
90 90
91 91
92 92
```

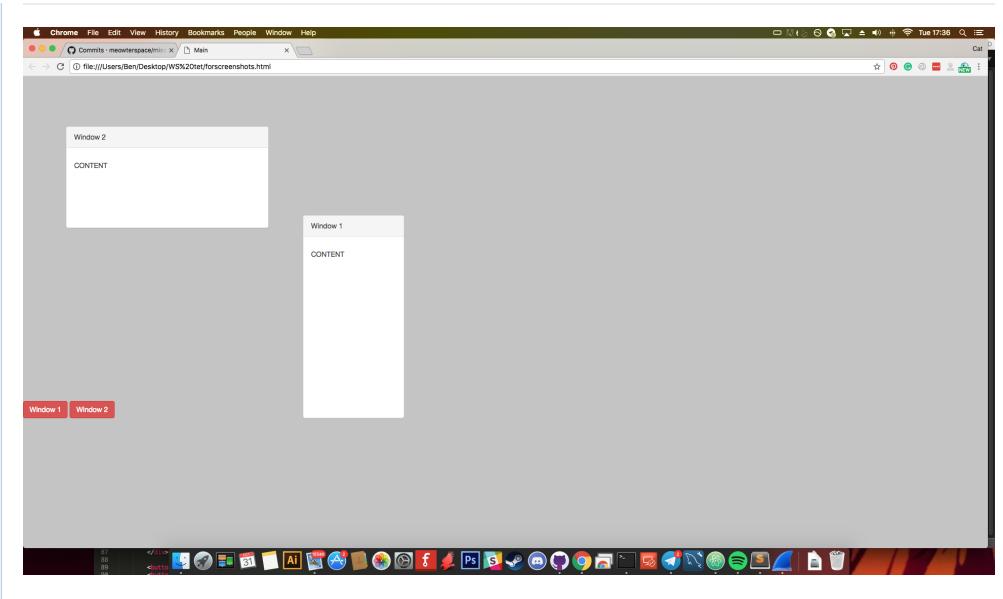
1 comment on commit f18e25d



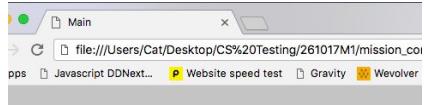
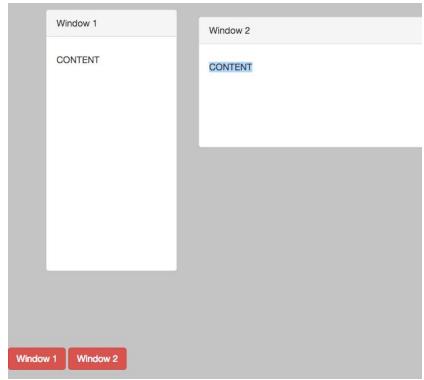
meowterspace commented on f18e25d on 31 Oct 2017

Owner

Preview



261017M3

Test	Pass/Fail	Description	Evidence
Check that the UI page loads	PASS	Page should load without error from a server. This is important to check after every change to make sure the page is unaffected. The JS error log should be empty	
In /ui.html, bootstrap styling should be applied across the entire page	PASS	Bootstrap styling should be obvious across every element of the page. This will make the user interface much more friendly	
In /ui.html, check modals can be loaded in any order	FAIL	<p>The modal should become visible in the order that the buttons were pressed. This is important for usability</p> <p>-----</p> <p>Modal number 2 can only appear after modal 1.</p>	

In /ui.html, check modals cannot be dragged outside the visible page boundaries	FAIL	The modal edge should bump against the page border edge so no part of the modal is outside the page. This stops a modal being lost or unrecoverable	 A screenshot of a browser window titled "Window 1". Inside the window, there is a modal dialog with the word "CONTENT" in its center. The modal is positioned such that its right edge is very close to the right edge of the browser window, illustrating that it is about to be dragged outside the visible page boundaries. The browser's address bar and tabs are visible at the bottom of the window.
---	-------------	---	---

Added navbar to replace buttons for windows

Browse files

I've added a bootstrap navbar, with each option opening a different window/modal. This should make it more user-friendly and compact, meaning we can get more in a small space.

Also added a new JS function to ensure the newly opened window floats ontop of anything else (high z-index), as previously it was displacing already existing windows which was a nuisance

` master

meowterspace committed on 26 Oct 2017

1 parent f18e25d commit 231d0f343797d821d335540f1fb5a828cace62dc

Showing 1 changed file with 47 additions and 47 deletions.

Unified Split

94 444444 ui.html

```

6 6      <!-- INCLUDE THE JQUERY LIBRARIES VIA CDN -->
7 7      <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
8 8      <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
9 9      -
10 10     -
11 11     <!-- INCLUDE THE BOOTSTRAP LIBRARIES VIA CDN -->
12 12     <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
13 13     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
14 14     -
15 15     <style type="text/css">
16 16     body {
17 17         background-color: #c4c4c4;
18 18         margin: 0px;
19 19         height: 100%;
20 20     }
21 21     .window {
22 22         border: 1px solid #000000;
23 23         background-color: #ffffff;
24 24         .handle {
25 25             background-color: #00ff00;
26 26             margin-top: 0em;
27 27         }
28 28         #win1 {
29 29             width: 200px;
30 30             height: 400px;
31 31             position: relative;
32 32             left: 5px; /* START POSITION ▲ ▼ */
33 33             top: 5px;
34 34             clear: both;
35 35             z-index: 1;
36 36             display: none;
37 37         }
38 38         #win2 {
39 39             width: 400px;
40 40             height: 200px;
41 41             position: relative;
42 42             left: 200px; /* START POSITION ▲ ▼ */
43 43             top: -400px;
44 44             clear: both;
45 45             z-index: 1;
46 46             display: none;
47 47     }
48 48     body {
49 49         background-color: #c4c4c4;
50 50         margin: 0px;
51 51         height: 100%;
52 52     }
53 53     #win1 {
54 54         width: 200px;
55 55         height: 400px;
56 56     }
57 57     #win2 {
58 58         width: 400px;
59 59         height: 200px;
60 60     }
61 61     .handle {
62 62         width: 10px;
63 63         height: 10px;
64 64         background-color: #00ff00;
65 65         position: absolute;
66 66         left: -5px;
67 67         top: -5px;
68 68         border-radius: 50%;
```

```

26 +         }
27 +         .panel {
28 +             position: absolute;
29 +             clear: both;
30 +             display: none;
31 +         }
32
33     </style>
34
35     /* This makes #winx draggable only by handle(class .handle)
36     with a change in opacity on event and ensures most recently dragged
37     has the highest z-index */
38
39     var zIndex = 1;
40
41     $(function() {
42         var zIndex = 1;
43
44         $("div[id*='win']").draggable({
45             handle: $(".handle"),
46             });
47
48         });
49
50     });
51
52     </script>
53
54     <script>
55
56
57     //var span = document.getElementsByClassName("close")[0];
58
59     /* Function for changing css attribute display to true.
60     On show increases z-index to be the highest of all windows
61     to prevent displacement */
62
63     function showModal(id) {
64         var win = document.getElementById(id);
65         $(win).css("z-index", ++zIndex);
66         $(win).show();
67     };
68
69
70     <body>
71
72         <div id="win1" class="panel panel-default">
73             <p class="handle panel-heading">Window 1</p>
74             <div class="panel-body">CONTENT</div>
75         </div>
76         <div id="win2" class="panel panel-default">
77             <p class="handle panel-heading">Window 2</p>
78             <div class="panel-body">CONTENT</div>
79         </div>
80
81         <nav class="navbar navbar-default">
82             <div class="container-fluid">
83                 <ul class="nav navbar-nav">
84                     <li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown" href="#">
85                         <ul class="dropdown-menu">
86                             <li><a href="javascript:showModal('win1')">T1</a></li>
87                             <li><a href="javascript:showModal('win2')">T2</a></li>
88                             <li><a href="#">T3</a></li>
89                         </ul>
90                     </li>
91                 </ul>
92             </div>
93         </nav>
94
95         <div id="win1" class="col-3 panel panel-default">
96             <p class="handle panel-heading">Window 1</p>
97             <div class="panel-body">CONTENT</div>
98         </div>
99
100         <button class="btn btn-danger" onclick="showModal('win1');">Window 1</button>
101         <button class="btn btn-danger" onclick="showModal('win2');">Window 2</button>
102
103         <div id="win2" class="col-2 panel panel-default">
104             <p class="handle panel-heading">Window 2</p>
105         </div>
106
107     </body>
108
109     <style>
110         .panel {
111             position: absolute;
112             clear: both;
113             display: none;
114         }
115
116         .panel .panel-body {
117             position: relative;
118             z-index: 1;
119         }
120
121         .panel .handle {
122             position: absolute;
123             top: 0;
124             left: 0;
125             width: 100px;
126             height: 100px;
127             background-color: #ccc;
128             border: 1px solid #ccc;
129             border-radius: 50%;
130             z-index: 2;
131         }
132
133         .panel .handle::before {
134             content: '';
135             position: absolute;
136             top: 50px;
137             left: 50px;
138             width: 0;
139             height: 0;
140             border-left: 10px solid transparent;
141             border-right: 10px solid transparent;
142             border-top: 20px solid #ccc;
143             z-index: 3;
144         }
145
146         .panel .panel-body::before {
147             content: '';
148             position: absolute;
149             top: 50px;
150             left: 50px;
151             width: 0;
152             height: 0;
153             border-left: 10px solid transparent;
154             border-right: 10px solid transparent;
155             border-top: 20px solid #ccc;
156             z-index: 3;
157         }
158
159         .panel .panel-body::after {
160             content: '';
161             position: absolute;
162             top: 50px;
163             left: 50px;
164             width: 0;
165             height: 0;
166             border-left: 10px solid transparent;
167             border-right: 10px solid transparent;
168             border-top: 20px solid #ccc;
169             z-index: 3;
170         }
171
172         .panel .panel-body::after {
173             content: '';
174             position: absolute;
175             top: 50px;
176             left: 50px;
177             width: 0;
178             height: 0;
179             border-left: 10px solid transparent;
180             border-right: 10px solid transparent;
181             border-top: 20px solid #ccc;
182             z-index: 3;
183         }
184
185         .panel .panel-body::after {
186             content: '';
187             position: absolute;
188             top: 50px;
189             left: 50px;
190             width: 0;
191             height: 0;
192             border-left: 10px solid transparent;
193             border-right: 10px solid transparent;
194             border-top: 20px solid #ccc;
195             z-index: 3;
196         }
197
198         .panel .panel-body::after {
199             content: '';
200             position: absolute;
201             top: 50px;
202             left: 50px;
203             width: 0;
204             height: 0;
205             border-left: 10px solid transparent;
206             border-right: 10px solid transparent;
207             border-top: 20px solid #ccc;
208             z-index: 3;
209         }
210
211         .panel .panel-body::after {
212             content: '';
213             position: absolute;
214             top: 50px;
215             left: 50px;
216             width: 0;
217             height: 0;
218             border-left: 10px solid transparent;
219             border-right: 10px solid transparent;
220             border-top: 20px solid #ccc;
221             z-index: 3;
222         }
223
224         .panel .panel-body::after {
225             content: '';
226             position: absolute;
227             top: 50px;
228             left: 50px;
229             width: 0;
230             height: 0;
231             border-left: 10px solid transparent;
232             border-right: 10px solid transparent;
233             border-top: 20px solid #ccc;
234             z-index: 3;
235         }
236
237         .panel .panel-body::after {
238             content: '';
239             position: absolute;
240             top: 50px;
241             left: 50px;
242             width: 0;
243             height: 0;
244             border-left: 10px solid transparent;
245             border-right: 10px solid transparent;
246             border-top: 20px solid #ccc;
247             z-index: 3;
248         }
249
250         .panel .panel-body::after {
251             content: '';
252             position: absolute;
253             top: 50px;
254             left: 50px;
255             width: 0;
256             height: 0;
257             border-left: 10px solid transparent;
258             border-right: 10px solid transparent;
259             border-top: 20px solid #ccc;
260             z-index: 3;
261         }
262
263         .panel .panel-body::after {
264             content: '';
265             position: absolute;
266             top: 50px;
267             left: 50px;
268             width: 0;
269             height: 0;
270             border-left: 10px solid transparent;
271             border-right: 10px solid transparent;
272             border-top: 20px solid #ccc;
273             z-index: 3;
274         }
275
276         .panel .panel-body::after {
277             content: '';
278             position: absolute;
279             top: 50px;
280             left: 50px;
281             width: 0;
282             height: 0;
283             border-left: 10px solid transparent;
284             border-right: 10px solid transparent;
285             border-top: 20px solid #ccc;
286             z-index: 3;
287         }
288
289         .panel .panel-body::after {
290             content: '';
291             position: absolute;
292             top: 50px;
293             left: 50px;
294             width: 0;
295             height: 0;
296             border-left: 10px solid transparent;
297             border-right: 10px solid transparent;
298             border-top: 20px solid #ccc;
299             z-index: 3;
300         }
301
302         .panel .panel-body::after {
303             content: '';
304             position: absolute;
305             top: 50px;
306             left: 50px;
307             width: 0;
308             height: 0;
309             border-left: 10px solid transparent;
310             border-right: 10px solid transparent;
311             border-top: 20px solid #ccc;
312             z-index: 3;
313         }
314
315         .panel .panel-body::after {
316             content: '';
317             position: absolute;
318             top: 50px;
319             left: 50px;
320             width: 0;
321             height: 0;
322             border-left: 10px solid transparent;
323             border-right: 10px solid transparent;
324             border-top: 20px solid #ccc;
325             z-index: 3;
326         }
327
328         .panel .panel-body::after {
329             content: '';
330             position: absolute;
331             top: 50px;
332             left: 50px;
333             width: 0;
334             height: 0;
335             border-left: 10px solid transparent;
336             border-right: 10px solid transparent;
337             border-top: 20px solid #ccc;
338             z-index: 3;
339         }
340
341         .panel .panel-body::after {
342             content: '';
343             position: absolute;
344             top: 50px;
345             left: 50px;
346             width: 0;
347             height: 0;
348             border-left: 10px solid transparent;
349             border-right: 10px solid transparent;
350             border-top: 20px solid #ccc;
351             z-index: 3;
352         }
353
354         .panel .panel-body::after {
355             content: '';
356             position: absolute;
357             top: 50px;
358             left: 50px;
359             width: 0;
360             height: 0;
361             border-left: 10px solid transparent;
362             border-right: 10px solid transparent;
363             border-top: 20px solid #ccc;
364             z-index: 3;
365         }
366
367         .panel .panel-body::after {
368             content: '';
369             position: absolute;
370             top: 50px;
371             left: 50px;
372             width: 0;
373             height: 0;
374             border-left: 10px solid transparent;
375             border-right: 10px solid transparent;
376             border-top: 20px solid #ccc;
377             z-index: 3;
378         }
379
380         .panel .panel-body::after {
381             content: '';
382             position: absolute;
383             top: 50px;
384             left: 50px;
385             width: 0;
386             height: 0;
387             border-left: 10px solid transparent;
388             border-right: 10px solid transparent;
389             border-top: 20px solid #ccc;
390             z-index: 3;
391         }
392
393         .panel .panel-body::after {
394             content: '';
395             position: absolute;
396             top: 50px;
397             left: 50px;
398             width: 0;
399             height: 0;
400             border-left: 10px solid transparent;
401             border-right: 10px solid transparent;
402             border-top: 20px solid #ccc;
403             z-index: 3;
404         }
405
406         .panel .panel-body::after {
407             content: '';
408             position: absolute;
409             top: 50px;
410             left: 50px;
411             width: 0;
412             height: 0;
413             border-left: 10px solid transparent;
414             border-right: 10px solid transparent;
415             border-top: 20px solid #ccc;
416             z-index: 3;
417         }
418
419         .panel .panel-body::after {
420             content: '';
421             position: absolute;
422             top: 50px;
423             left: 50px;
424             width: 0;
425             height: 0;
426             border-left: 10px solid transparent;
427             border-right: 10px solid transparent;
428             border-top: 20px solid #ccc;
429             z-index: 3;
430         }
431
432         .panel .panel-body::after {
433             content: '';
434             position: absolute;
435             top: 50px;
436             left: 50px;
437             width: 0;
438             height: 0;
439             border-left: 10px solid transparent;
440             border-right: 10px solid transparent;
441             border-top: 20px solid #ccc;
442             z-index: 3;
443         }
444
445         .panel .panel-body::after {
446             content: '';
447             position: absolute;
448             top: 50px;
449             left: 50px;
450             width: 0;
451             height: 0;
452             border-left: 10px solid transparent;
453             border-right: 10px solid transparent;
454             border-top: 20px solid #ccc;
455             z-index: 3;
456         }
457
458         .panel .panel-body::after {
459             content: '';
460             position: absolute;
461             top: 50px;
462             left: 50px;
463             width: 0;
464             height: 0;
465             border-left: 10px solid transparent;
466             border-right: 10px solid transparent;
467             border-top: 20px solid #ccc;
468             z-index: 3;
469         }
470
471         .panel .panel-body::after {
472             content: '';
473             position: absolute;
474             top: 50px;
475             left: 50px;
476             width: 0;
477             height: 0;
478             border-left: 10px solid transparent;
479             border-right: 10px solid transparent;
480             border-top: 20px solid #ccc;
481             z-index: 3;
482         }
483
484         .panel .panel-body::after {
485             content: '';
486             position: absolute;
487             top: 50px;
488             left: 50px;
489             width: 0;
490             height: 0;
491             border-left: 10px solid transparent;
492             border-right: 10px solid transparent;
493             border-top: 20px solid #ccc;
494             z-index: 3;
495         }
496
497         .panel .panel-body::after {
498             content: '';
499             position: absolute;
500             top: 50px;
501             left: 50px;
502             width: 0;
503             height: 0;
504             border-left: 10px solid transparent;
505             border-right: 10px solid transparent;
506             border-top: 20px solid #ccc;
507             z-index: 3;
508         }
509
510         .panel .panel-body::after {
511             content: '';
512             position: absolute;
513             top: 50px;
514             left: 50px;
515             width: 0;
516             height: 0;
517             border-left: 10px solid transparent;
518             border-right: 10px solid transparent;
519             border-top: 20px solid #ccc;
520             z-index: 3;
521         }
522
523         .panel .panel-body::after {
524             content: '';
525             position: absolute;
526             top: 50px;
527             left: 50px;
528             width: 0;
529             height: 0;
530             border-left: 10px solid transparent;
531             border-right: 10px solid transparent;
532             border-top: 20px solid #ccc;
533             z-index: 3;
534         }
535
536         .panel .panel-body::after {
537             content: '';
538             position: absolute;
539             top: 50px;
540             left: 50px;
541             width: 0;
542             height: 0;
543             border-left: 10px solid transparent;
544             border-right: 10px solid transparent;
545             border-top: 20px solid #ccc;
546             z-index: 3;
547         }
548
549         .panel .panel-body::after {
550             content: '';
551             position: absolute;
552             top: 50px;
553             left: 50px;
554             width: 0;
555             height: 0;
556             border-left: 10px solid transparent;
557             border-right: 10px solid transparent;
558             border-top: 20px solid #ccc;
559             z-index: 3;
560         }
561
562         .panel .panel-body::after {
563             content: '';
564             position: absolute;
565             top: 50px;
566             left: 50px;
567             width: 0;
568             height: 0;
569             border-left: 10px solid transparent;
570             border-right: 10px solid transparent;
571             border-top: 20px solid #ccc;
572             z-index: 3;
573         }
574
575         .panel .panel-body::after {
576             content: '';
577             position: absolute;
578             top: 50px;
579             left: 50px;
580             width: 0;
581             height: 0;
582             border-left: 10px solid transparent;
583             border-right: 10px solid transparent;
584             border-top: 20px solid #ccc;
585             z-index: 3;
586         }
587
588         .panel .panel-body::after {
589             content: '';
590             position: absolute;
591             top: 50px;
592             left: 50px;
593             width: 0;
594             height: 0;
595             border-left: 10px solid transparent;
596             border-right: 10px solid transparent;
597             border-top: 20px solid #ccc;
598             z-index: 3;
599         }
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1220
1221
1222
1223
1224

```

```
89 | +             <div class="panel-body">CONTENT</div>
90 | +         </div>
91 |
92 | -         </body>
92 | +</body>
```

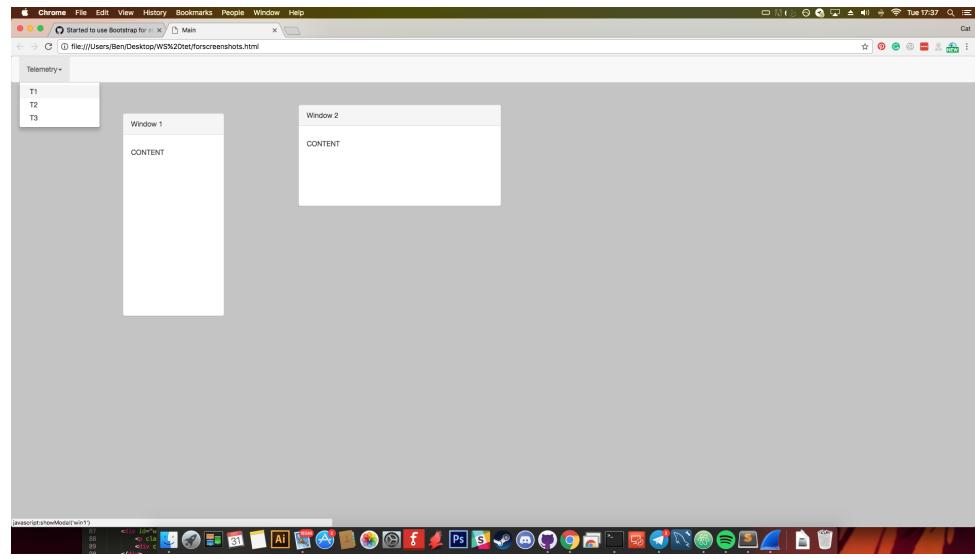
1 comment on commit 231d0f3



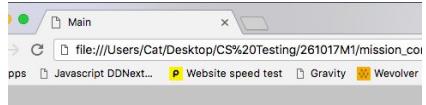
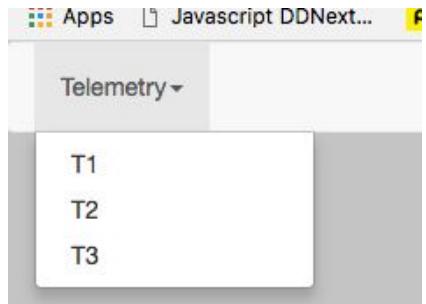
meowterspace commented on 231d0f3 on 31 Oct 2017

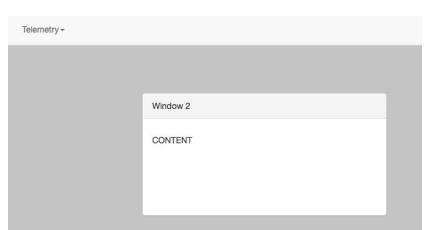
Owner

Preview



261017M4

Test	Pass/Fail	Description	Evidence
Check that the UI page loads	PASS	Page should load without error from a server. This is important to check after every change to make sure the page is unaffected. The JS error log should be empty	
In /ui.html, check that a navigation bar can be added	PASS	The navigation bar should appear at the top of the page as per design	
Nav-bar drop down menu	PASS	When a button is clicked on the nav-bar, a drop down menu should appear with more buttons	

Nav-bar buttons show modals	PASS	The Nav-bar button should show the corresponding modal when clicked	
In /ui.html, check modals can be loaded in any order	PASS	The modal should become visible in the order that the buttons were pressed. This is important for usability	

In /ui.html, check modals cannot be dragged outside the visible page boundaries	FAIL	The modal edge should bump against the page border edge so no part of the modal is outside the page. This stops a modal being lost or unrecoverable	
---	-------------	---	---

Added a close/hide modal function

[Browse files](#)

Very basic functionality added to hide the modal.
Function called onclick by bootstrap button in modal heading

by master

 meowterspace committed on 26 Oct 2017

1 parent 231d0f3 commit 42b375431a2c38a8c8b23549ff36346e28f225fc

Showing 1 changed file with 16 additions and 2 deletions.

[Unified](#) [Split](#)

18 ████ ui.html

```
59 59             $(win).show();  
60 60         };  
61 61  
62 62     /* Function for hiding the modal */  
63 63     function hideModal(id) {  
64 64         var win = document.getElementById(id);  
65 65         $(win).hide();  
66 66     }  
67 67  
68 68         </script>  
69 69  
70 70     </head>  
86 86     </nav>  
87 87  
88 88     <div id="win1" class="col-3 panel panel-default">  
83 83         <p class="handle panel-heading">Window 1</p>  
89 89         <div class="handle panel-heading">Window 1  
90 90             <button type="button" class="close" aria-label="Close" onclick="hideModal('win1')">  
91 91                 <span aria-hidden="true">&times;</span>  
92 92             </button>  
93 93         </div>  
94 94         <div class="panel-body">CONTENT</div>  
95 95     </div>  
96 96  
97 97     <div id="win2" class="col-2 panel panel-default">  
88 88         <p class="handle panel-heading">Window 2</p>  
98 98         <div class="handle panel-heading">Window 2  
99 99             <button type="button" class="close" aria-label="Close" onclick="hideModal('win2')">  
100 100                 <span aria-hidden="true">&times;</span>  
101 101             </button>  
102 102         </div>  
103 103         <div class="panel-body">CONTENT</div>  
104 104     </div>  
91 105
```

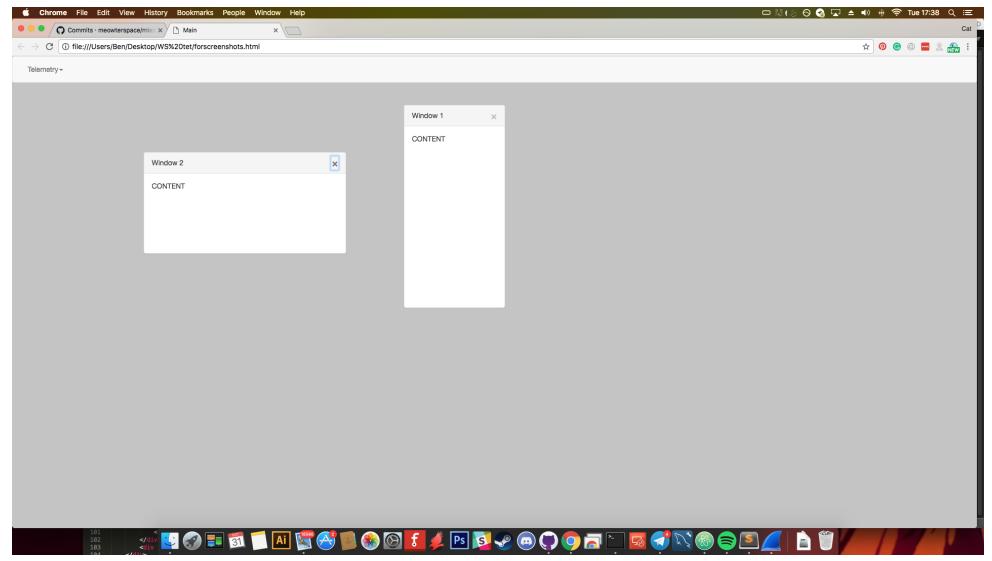
1 comment on commit 42b3754



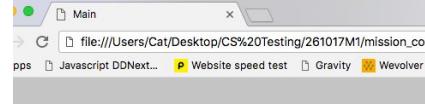
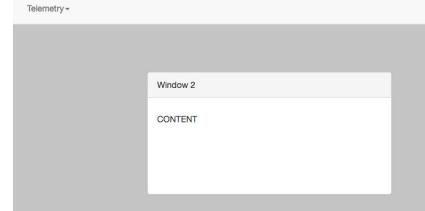
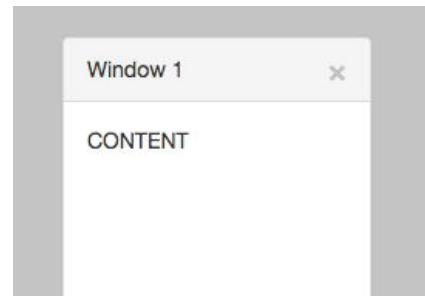
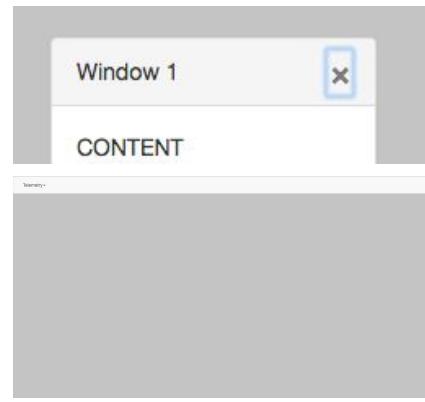
meowterspace commented on 42b3754 on 31 Oct 2017

Owner

Preview



261017M5

Test	Pass/Fail	Description	Evidence
Check that the UI page loads	PASS	Page should load without error from a server. This is important to check after every change to make sure the page is unaffected. The JS error log should be empty	
In /ui.html, check modals can be loaded in any order	PASS	The modal should become visible in the order that the buttons were pressed. This is important for usability	
Exit button in handle	PASS	An [x] Exit button should appear in the top right corner of the modal handle	
Exit button closes modal	PASS	When the [x] button is clicked that modal and ONLY that modal should become hidden	

In /ui.html, check modals cannot be dragged outside the visible page boundaries	FAIL	The modal edge should bump against the page border edge so no part of the modal is outside the page. This stops a modal being lost or unrecoverable	
---	-------------	---	---

Added boundaries for windows

[Browse files](#)

I've made it so a user cannot drag a window off the screen, to prevent confusion or loss of window.

I've also disabled the scroll feature.

by master

meowterspace committed on 26 Oct 2017

1 parent 42b3754 commit 2aa93009b4537a76174e847b178868c09b5966d6

Showing 1 changed file with 10 additions and 9 deletions.

[Unified](#) [Split](#)

19  ui.html

```
34 34
35 35             <script>
36 36
37 37 +             /* This makes #winx draggable only by handle(class .handle)
38 38             with a change in opacity on event and ensures most recently dragged
39 39             has the highest z-index */
40 40             var zIndex = 1;
41 41             $(function() {
42 42
43 43                 $("div[id*='win']").draggable({
44 44                     containment: "window",
45 45 +                     scroll: false,
46 46 +                     handle: $(".handle"),
47 47                     opacity: 0.8,
48 48                     start: function(event, ui) {
49 49
50 50                 });
51 51
52 52
53 53             /* Function for changing css attribute display to true.
54 54             - On show increases z-index to be the highest of all windows
55 55             - to prevent displacement */
56 56             On show increases z-index to be the highest of all windows
57 57             - to prevent displacement */
58 58             + to prevent displacement */
59 59             function showModal(id) {
60 60                 var win = document.getElementById(id);
61 61                 $(win).css("z-index", ++zIndex);
62 62             function hideModal(id) {
63 63                 var win = document.getElementById(id);
64 64                 $(win).hide();
65 65
66 66             }
67 67
68 68             });
69 69
70 70             +;
71 71
72 72             </script>
73 73
74 74         </head>
75 75
76 76         <body>
77 77
78 78             <nav class="navbar navbar-default">
79 79                 <div class="container-fluid">
80 80                     <ul class="nav navbar-nav">
81 81                     </ul>
82 82                 </div>
83 83             </nav>
84 84
85 85             <div id="win1" class="col-3 panel panel-default">
86 86                 <div class="handle panel-heading">Window 1
87 87                     <button type="button" class="close" aria-label="Close" onclick="hideModal('win1')">
88 88                         <span aria-hidden="true">&times;</span>
89 89                         <span aria-hidden="true">&times;</span>
90 90                     </button>
91 91
92 92
93 93 +                 <span aria-hidden="true">&times;</span>
94 94             </div>
```

```
93  95          </div>
94  96          <div class="panel-body">CONTENT</div>
95  97      </div>
96
97      - <div id="win2" class="col-2 panel panel-default">
98  99      + <div id="win2" class="col-2 panel panel-danger">
99 100          <div class="handle panel-heading">Window 2
100 101          <button type="button" class="close" aria-label="Close" onclick="hideModal('win2')">
101 102          - <span aria-hidden="true">&times;</span>
102 103          + <span aria-hidden="true">&times;</span>
103 104          </button>
104 105      </div>
105 106      <div class="panel-body">CONTENT</div>
106 107  </div>
107
108      - </body>
```

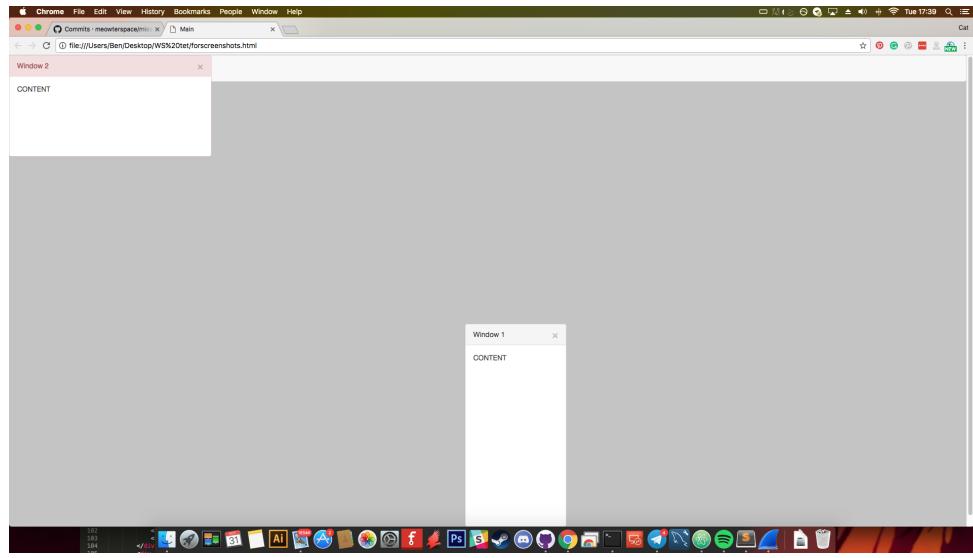
1 comment on commit 2aa9300



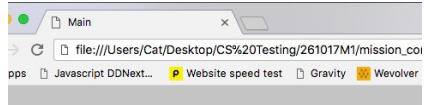
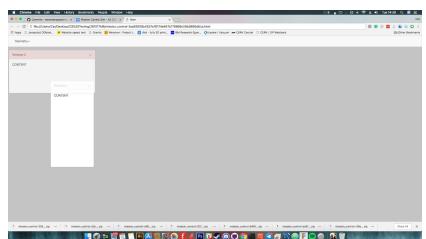
meowterspace commented on 2aa9300 on 31 Oct 2017

Owner

Preview



261017M6

Test	Pass/Fail	Description	Evidence
Check that the UI page loads	PASS	Page should load without error from a server. This is important to check after every change to make sure the page is unaffected. The JS error log should be empty	
In /ui.html, check modals cannot be dragged outside the visible page boundaries	PASS	The modal edge should bump against the page border edge so no part of the modal is outside the page. This stops a modal being lost or unrecoverable	

Started development of web sockets

1 experiments

[Browse files](#)

 meowterspace committed on 31 Oct 2017

1 parent 2aa9300 commit 8756a4ab1d63899e764679f647d24ca7e58502b5

 Showing 4 changed files with 60 additions and 108 deletions.

[Unified](#) [Split](#)

32 Websockets test/index.html

```
... ... @@ -0,0 +1,32 @@
1 +<html>
2 +<head>
3 +<title>Chat Room</title>
4 +<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/1.4.8/socket.io.min.js"></script>
5 +<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
6 +</head>
7 +<body>
8 +<script type="text/javascript">
9 +$(document).ready(function() {
10 +
11 +    var socket = io.connect('http://127.0.0.1:5000');
12 +
13 +    socket.on('connect', function() {
14 +        socket.send('User has connected!');
15 +    });
16 +
17 +    socket.on('message', function(msg) {
18 +        $("#messages").append('<li>' + msg + '</li>');
19 +        socket.send(msg + ' received')
20 +        console.log(msg);
21 +        alert(msg);
22 +    });
23 +
24 +
25 +
26 +});
27 +</script>
28 +<ul id="messages"></ul>
29 +<input type="text" id="myMessage">
30 +<button id="sendbutton">Send</button>
31 +</body>
32 +</html>
```

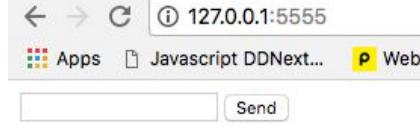
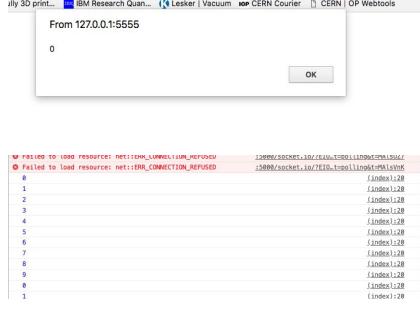
27 Websockets test/server.py

```
... ... @@ -0,0 +1,27 @@
1 +import time
2 +from flask import Flask
3 +from flask_socketio import SocketIO, emit, send
4 +from threading import Thread
5 +
6 +app = Flask(__name__)
7 +app.config['SECRET_KEY'] = 'secret'
8 +socketio = SocketIO(app)
9 +
10 +
11 +def bg_emit(count):
12 +    send(count)
13 +
14 +@socketio.on('message')
15 +def handle_message(message):
16 +    print(message)
17 +
18 +    for i in range(10):
```

```
19      +         bg_emit(i)
20      +         print(i)
21      +         time.sleep(1)
22      +
23      +
24  +if __name__ == '__main__':
25      +     socketio.run(app)
26      +
27      +
```

0 comments on commit [8756a4a](#)

311017E1

Test	Pass/Fail	Description	Evidence
Check that a flask server can be started	PASS	Look at the server terminal. The server should quote a debugger pin and a server address. Important to check the server works before continuing.	
Check that the server can be accessed by correct IP and Port	PASS	The address should load the defined index page without error. This is to make sure the server can be accessed remotely by the client	
Check that the server receives a websocket message when the client page loads	PASS	Look at server terminal. Printout 'User has connected' should appear. This is to ensure that the server can receive a trigger to start sending data	
Check that the server sends the client messages in response to each received	PASS	Observe JS readout and check messages are received via websockets to make sure data can be communicated between client and server.	
Check that the server sends and the client receives messages in sync	FAIL	Watch both JS client console and server terminal to see if the messages (0-9) are in sync. This will be important for real time feedback in the game.	

Check that client sends back message to server	PASS	Watch server terminal. Message should be received after every message sent. This makes sure the process is a continuous cycle	<pre>,</pre> <pre>8</pre> <pre>9</pre> <pre>1 received</pre> <pre>0</pre> <pre>1</pre> <pre>2</pre> <pre>3</pre> <pre>4</pre> <pre>5</pre> <pre>6</pre> <pre>7</pre> <pre>8</pre> <pre>9</pre> <pre>2 received</pre> <pre>0</pre> <pre>1</pre> <pre>2</pre> <pre>3</pre>
--	-------------	--	--



Fixed problem, client now receives constant stream of data

Browse files

see:

#1

💡 experiments

 meowterspace committed on 31 Oct 2017

1 parent 8756a4a commit 9b001932e87a8c548009b440aed3219c8edb0006

 Showing 2 changed files with 8 additions and 13 deletions.

Unified | Split

9  Websockets test/index.html

```
8 8 <script type="text/javascript">
9 9 $(document).ready(function() {
10 10
11 11 - var socket = io.connect('http://127.0.0.1:5000');
12 12 + var socket = io.connect('http://127.0.0.1:5000');
13 13 - socket.on('connect', function() {
14 14 -     socket.send('User has connected!');
15 15 - });
16 16 + socket.on('connect', function() {
17 17 +     socket.send('User has connected!');
18 18 + });
19 19
20 20     socket.on('message', function(msg) {
21 21         //$("#messages").append('<li>' + msg + '</li>');
22 22         socket.send(msg + ' received')
23 23         console.log(msg);
24 24 -         alert(msg);
25 25     });
26 26 }
```

12 Websockets test/server.py

```
8 8     socketio = SocketIO(app)
9 9
10 10
11 11     -def bg_emit(count):
12 12         -    send(count)
13 13         -
14 14     +count = 0
15 15     @socketio.on('message')
16 16     def handle_message(message):
17 17         -
18 18             -    for i in range(10):
19 19                 -        bg_emit(i)
20 20                 -        print(i)
21 21                 -        time.sleep(1)
22 22
23 23
24 24     +    count += 1
25 25     +    send(count)
26 26
27 27
28 28     if __name__ == '__main__':
29 29         socketio.run(app)
30 30
31 31     +
```

0 comments on commit 9b00193

Fixed & write to document

[Browse files](#)

Last commit was wrong.

This one fixes following the issue[1] noted.

Also I've written to the document to change the time in a nice way :D
Will try sending JSONs next for more data!

↳ experiments

 meowterspace committed on 31 Oct 2017

1 parent 9b00193 commit 8761984746545e407ee1fd6187d49bd1e1ff3054

Showing 2 changed files with 6 additions and 11 deletions.

[Unified](#) [Split](#)

6  Websockets test/index.html

```
15 15      });
16 16
17 17      socket.on('message', function(msg) {
18 18      - //$("#messages").append('<li>' + msg + '</li>');
19 19      + document.getElementById("time").innerHTML = msg;
20 20      socket.send(msg + ' received')
21 21      console.log(msg);
22 22
23 23      });
24 24
25 25      });
26 26      </script>
27 27      -<ul id="messages"></ul>
28 28      -<input type="text" id="myMessage">
29 29      -<button id="sendbutton">Send</button>
30 30      +<h1>Time: <h1><h1 id="time"></h1>
31 31      </body>
32 32      </html>
```

11  Websockets test/server.py

```
...
...
@@ -1,4 +1,5 @@
1     import time
2     +import datetime
3     from flask import Flask
4     from flask_socketio import SocketIO, emit, send
5     from threading import Thread
6     socketio = SocketIO(app)
7
8
9
10
11     -count = 0
12     @socketio.on('message')
13     def handle_message(message):
14         print(message)
15         -     count += 1
16         -     send(count)
17         +     x = str(datetime.datetime.now())
18         +     send(x)
19
20     if __name__ == '__main__':
21         -     socketio.run(app)
22
23
24     +     socketio.run(app) ↗
```

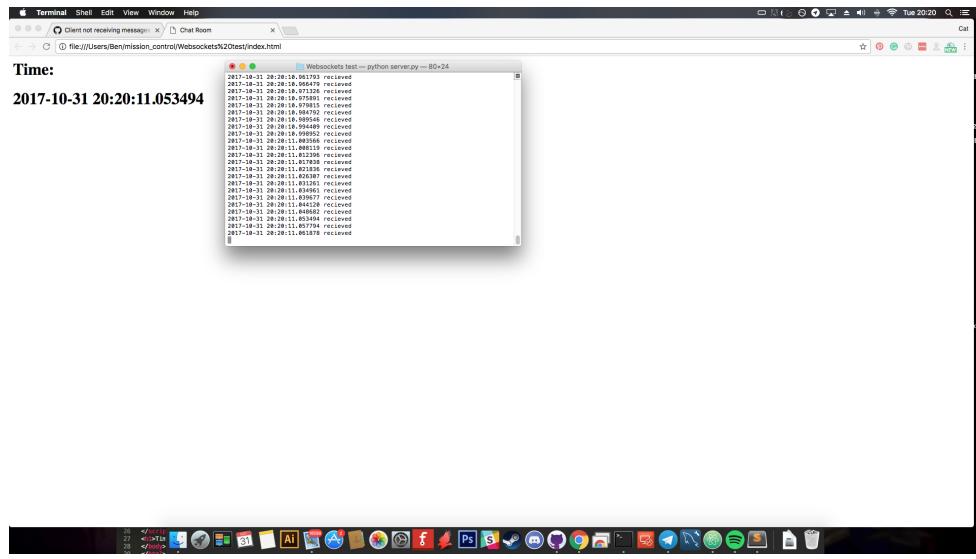
1 comment on commit 8761984



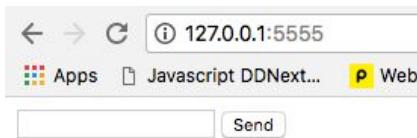
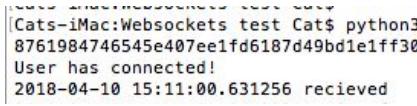
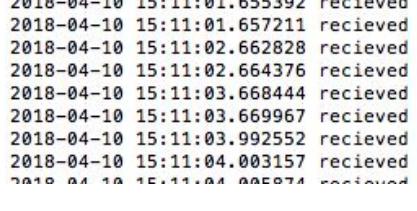
meowterspace commented on 8761984 on 31 Oct 2017

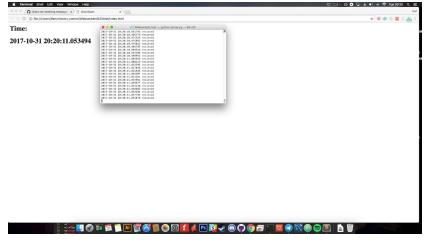
Owner

Preview



311017E3

Test	Pass/Fail	Description	Evidence
Check that a flask server can be started	PASS	Look at the server terminal. The server should quote a debugger pin and a server address. Important to check the server works before continuing.	
Check that the server can be accessed by correct IP and Port	PASS	The address should load the defined index page without error. This is to make sure the server can be accessed remotely by the client	
Check that the server receives a websocket message when the client page loads	PASS	Look at server terminal. Printout 'User has connected' should appear. This is to ensure that the server can receive a trigger to start sending data	
Check that the server can send longer messages	PASS	Check the server terminal. Messages should still be sending as before. This verifies that strings can be send as well as integers	
Check that the client can receive longer messages	PASS	The JS client console should show messages being received as before. This verifies that strings can be received as well as integers	

Check that the server sends and the client receives messages in sync	PASS	Watch both JS client console and server terminal to see if the messages (0-9) are in sync. This will be important for real time feedback in the game.	
--	-------------	---	---

Browse files

Sends data over JSONs

I've set it up to send data in JSON format. This will be far easier when sending big lots of data later for the launch simulation.

↳ experiments

meowterspace committed on 31 Oct 2017

1 parent 8761984 commit d34fbbed442ab1c7e514d11d3d85bffd7555a23d1

Showing 2 changed files with 17 additions and 6 deletions.

Unified Split

9 Websockets test/index.html

```

15  15  });
16  16
17  17  socket.on('message', function(msg) {
18  18  -  document.getElementById("time").innerHTML = msg;
19  19  -  socket.send(msg+' recieived')
20  20  +
21  21  +  document.getElementById('time').innerHTML = 'Time ('+msg.zone+'): '+msg.time;
22  22  +  document.getElementById('serv').innerHTML = 'Server runtime: '+msg.serv;
23  23  +  socket.send(msg+' recieived');
24  24  console.log(msg);
25  25  });
26  26
27  27  });
28  28  </script>
29  29  -<h1>Time: <h1 id="time"></h1>
30  30  +<h1 id="time"></h1>
31  31  +<h1 id="serv"></h1>
32  32  </body>
</html>

```

14 Websockets test/server.py

```

...
...
@@ -1,4 +1,4 @@
1  -import time
1  +import json
2  2  import datetime
3  3  from flask import Flask
4  4  from flask_socketio import SocketIO, emit, send
8  8  app.config['SECRET_KEY'] = 'secret'
9  9  socketio = SocketIO(app)
10
11 +data = {
12  +  'time' : '',    #current Time
13  +  'zone' : 'GMT', #timezone
14  +  'serv' : ''     #time the server's been acrive
15  +}
16
17 +start = datetime.datetime.now()
18
19 @socketio.on('message')
20 def handle_message(message):
21     print(message)
22     -  x = str(datetime.datetime.now())
23     -  send(x)
24     +  data['time'] = str(datetime.datetime.now())
25     +  data['serv'] = str(datetime.datetime.now()-start)
26     +  send(data)
27

```

1 comment on commit d34fbbed



meowterspace commented on d34fbed on 31 Oct 2017

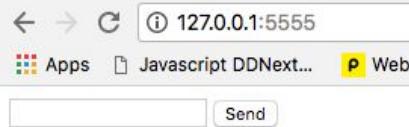
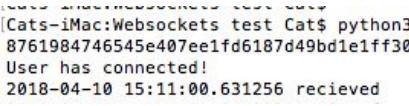
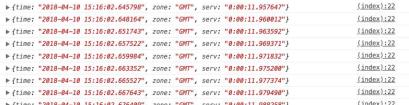
Owner

Preview

Time (GMT): 2017-10-31 20:58:25.990319

Server runtime: 0:00:26.977544

311017E4

Test	Pass/Fail	Description	Evidence
Check that a flask server can be started	PASS	Look at the server terminal. The server should quote a debugger pin and a server address. Important to check the server works before continuing.	
Check that the server can be accessed by correct IP and Port	PASS	The address should load the defined index page without error. This is to make sure the server can be accessed remotely by the client	
Check that the server receives a websocket message when the client page loads	PASS	Look at server terminal. Printout 'User has connected' should appear. This is to ensure that the server can receive a trigger to start sending data	
Check that the server can send objects over websockets	PASS	Check server terminal. Objects should be sent. This is important as it means multiple bits of data can be sent in one packet.	
Check that the client can receive objects	PASS	Check JS client console. Objects should be printed and obvious. This is important because it means the client can receive multiple pieces of information at once	

Air Pressure Calculation Test

Browse files

I've quickly written a function utilising the barometric formula to take pressure at sea level, ambient temperature and height to calculate the pressure at that height.

To test I ran this through matplotlib to get a graph, which confirmed the code works

💡 experiments

 meowterspace committed on 28 Nov 2017

1 parent d34fbbed commit 64b3d18fd8b15b734c49d4b6724728853df857b3

 Showing **1 changed file** with **20 additions** and **0 deletions**.

Unified Split

20 Computation Engine/air_pressure.py

```
...
@@ -0,0 +1,20 @@
1 +import numpy as np
2 +import matplotlib.pyplot as mpl
3 +
4 +#
5 +
6 +def atmos_pressure(P0, T, h):          # P0 = Pressure at sea Level
7 +    P = P0*np.exp((0.02896 * 9.807)/(8.3143*T))*h  # T = Temperature of ambient air (K)
8 +    return P
9 +
10 +
11 +
12 +heights = []
13 +data = []
14 +
15 +for i in range (100000):
16 +    heights.append(i)
17 +    data.append(atmos_pressure(101325, 288.15, i))
18 +
19 +mpl.plot(heights, data)
20 +mpl.show() Ø
```

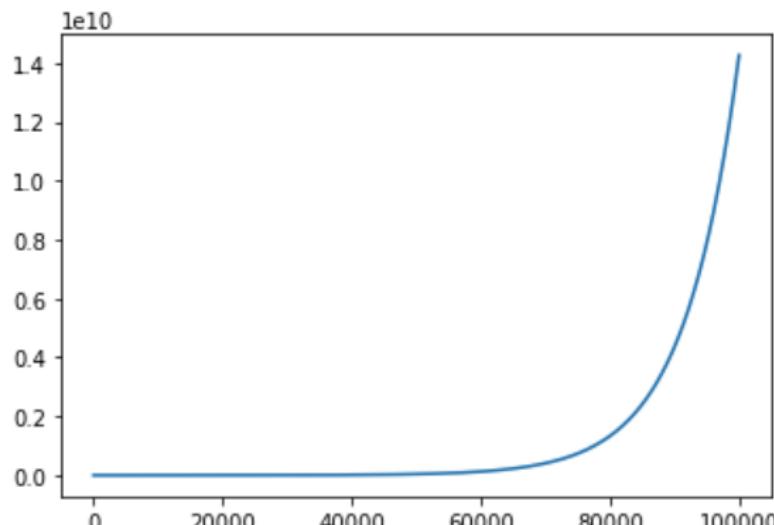
1 comment on commit 64b3d18

A small, round cartoon character with large, expressive eyes and pink hair styled in a bun with a white bow. The character is holding a small, dark gray or black cat in its arms. The background is a soft pink gradient.

meowterspace commented on 64b3d18 on 28 Nov 2017

Owner

Results from matplotlib



height/m vs pressure/bar

281117E1

Test	Pass/Fail	Description	Evidence
Check that the air pressure calculation functions correctly	PASS	The air pressure calculation should when plotted against height match a logarithmic curve like so:	<p>This graph is correct however the axis are reversed so the graph shows an inverse log graph (exponential).</p> <pre> import numpy as np import matplotlib.pyplot as plt from . import sure def altitude_vs_pressure(): alt = np.logspace(0, 5, 1000) pres = 100000 * np.exp(-alt) plt.loglog(alt, pres) plt.title('Altitude vs. Pressure') plt.xlabel('Altitude (m)') plt.ylabel('Pressure (Pa)') plt.grid(True) plt.show() altitude_vs_pressure() </pre>

Thrust calculation

Browse files

I've written a function to calculate thrust at a given point. The only real variable in this will be the ambient atmospheric pressure, which I can calculate at every point using my barometric formula

💡 experiments

 meowterspace committed on 28 Nov 2017

1 parent 64b3d18 commit 56538ef7eaf8feec1661f809c9aa2b3cd14bbda6

 Showing **1 changed file** with **9 additions** and **0 deletions**.

Unified Split

9  Computation Engine/thrust.py

0 comments on commit 56538ef

281117E2

Test	Pass/Fail	Description	Evidence
Check that the thrust calculation functions correctly	PASS	<p>The the thrust calculation should match the results of this calculation:</p> <div style="background-color: #f0f0f0; padding: 10px;"> <p>PROBLEM 1.1 A spacecraf's engine expels mass at a rate of 30 kg/s with an exhaust velocity of 3100 m/s. The pressure at the nozzle exit is 5 kPa and the exit area is 0.7 m². What is the thrust of the engine in a vacuum?</p> <p>SOLUTION, Given: $q = 30 \text{ kg/s}$ $V_e = 3,100 \text{ m/s}$ $A_e = 0.7 \text{ m}^2$ $P_e = 5 \text{ kPa} = 5,000 \text{ N/m}^2$ $P_\infty = 0$ Equation (1.6), $F = q \times V_e + (P_\infty - P_e) \times A_e$ $F = 30 \times 3,100 + (5,000 - 0) \times 0.7$ $F = 96,500 \text{ N}$</p> </div> <p>Where $F = 96500$</p>	<pre>----->>> import thrust >>> thrust.thrust(30, 3100, 5000, 0, 0.7) 96500.0 >>> </pre>



Drag equation

Browse files

I've written a function to calculate atmospheric drag



 meowterspace committed on 28 Nov 2017

1 parent 56538ef commit a6aac875866f7d98419b7c47da2cee8ce020a576

 Showing **1 changed file** with **9 additions** and **0 deletions**.

Unified Split

9 Computation Engine/atmospheric_drag.py

0 comments on commit a6aac87

281117E3

Test	Pass/Fail	Description	Evidence
Check that the atmospheric drag calculation function correctly	PASS	The atmospheric drag equation should return a sensible positive result that is lower than the value of F from above test. This proves that the atmos drag function works	<pre>>>> import atmospheric_drag as ad >>> ad.drag(0.4, 0.286, 283, 50, 0.42, 1.05) 21817.06 >>></pre>



General Gravitation Attraction calculation

Browse files

I've used the standard GM_1M_2/R^2 equation to calculate the objects attraction towards the earth or any other body

💡 experiments



 meowterspace committed on 28 Nov 2017

1 parent a6aac87 commit 51dcc2cd9a31f165601add7ed3fa234c192eb0ed

 Showing 1 changed file with 9 additions and 0 deletions.

Unified Split

9 Computation Engine/Gravitational Attraction.py

0 comments on commit 51dcc2c

281117E4

Test	Pass/Fail	Description	Evidence
Check that the gravitational attraction calculations work	PASS	The equation should return a force of ~4E-45. This proves that the gravitational attraction function works	<pre>[>>> import GravitationalAttraction as ga [>>> ga.gravity(6E24, 10, 1E30) 4.004447999999993e-45</pre>

Put functions together & tested

Browse files

I've started to put the functions in one file to make it easier and more understandable for me to access at a later date. This way I only have to import one file rather than many.

I've written a test script at the bottom to calculate the overall force in one dimension of a rocket - using the other functions.

I plotted atmospheric drag vs height to check it was correct

💡 experiments

 meowterspace committed on 30 Nov 2017

1 parent 51dcc2c commit 69dcf80c1c716091bfc29216bc9f79ab8ebf2236

 Showing 2 changed files with 50 additions and 2 deletions.

Unified Split

5 Computation Engine/atmospheric_drag.py

47 Computation Engine/compute.py

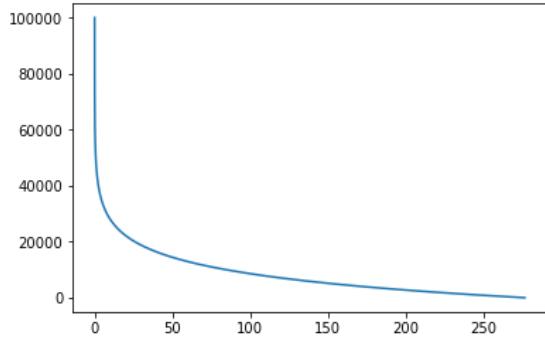
```
35 +
36 +     F = F - (gravity(5.927e24, 100, (6.371e6 + h)))
37 +     r = drag(P, 288.15, 30, 0.5, 1)
38 +     F = F - drag(P, 288.15, 30, 0.5, 1)
39 +
40 +     print(r, F, h)
41 +     data.append(r)
42 +     heights.append(h)
43 +
44 +mpl.plot(data, heights)
45 +mpl.show()
46 +
47 +
```

1 comment on commit 69dcf80



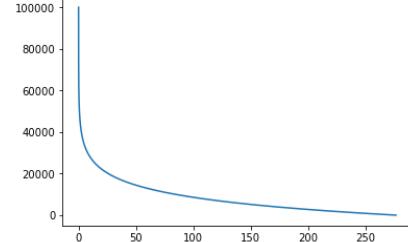
meowterspace commented on 69dcf80 on 30 Nov 2017

Owner



Height/m Vs Atmospheric Drag/N

301117E1

Test	Pass/Fail	Description	Evidence
Check that all major mathematical functions integrate together	PASS	The function when plotted should show a steep drop and then a logarithmic gradual decline. This is to check that all the functions work together	

[Browse files](#)

Started development of computational engine

I've started development of a computational engine to get a hang of the coordinate system. The coordinates use XYZ in meters, and the angles are Euler angles in terms of Yaw, Pitch Roll.

I've used an object orientated approach to make it easier to use and reference objects (such as rocket & planet) later in the code. I will move bits of code around later to make it neater.

At the bottom (# In[59]) I've written some code to run plot the path of the rocket in 3 dimensions. I've been changing the angle of the rocket to see if the coordinate system works with the euler angles.

I've concluded the coordinate system works fine for in atmosphere movements

↳ experiments

 meowterspace committed on 7 Dec 2017

1 parent 69dcf80 commit e2eab4b52a2ef6d9803904a960e4b7523155b696

Showing 2 changed files with 118 additions and 0 deletions.

[Unified](#) [Split](#)

110  Computation Engine/Compute_engine_test.py

```
...  ... @@ -0,0 +1,110 @@
1 +
2 +## coding: utf-8
3 +
4 +## In[1]:
5 +
6 +import numpy as np
7 +
8 +
9 +## In[2]:
10 +
11 +def drag(P, T, Vg, Cd, A): # P = Pressure /PA, T = temp(k), Vg = flow velocity of gas
12 +
13 +    F = 0.5 * ((P / (286 * T)) * (Vg ** 2) * Cd * A) # Cd = Coefficient of drag, A = area in drag
14 +    return F
15 +
16 +
17 +def gravity(M, m, r): # P = Pressure, T = temp(k), Vg = flow velocity of gas
18 +    F = (6.67408e-11 * M * m) / (r ** 2) # Cd = Coefficient of drag, A = area in drag
19 +    return F
20 +
21 +
22 +def thrust(q, Ve, Pe, Pa, Ae): # q = rate of ejected mass flow, Ve = exhaust gas ejection speed
23 +    F = q * Ve + (Pe - Pa) * Ae # Pe = pressure of exhaust gasses, Pa = pressure of ambient atmosphere
24 +    return F # Ae = area of exit
25 +
26 +
27 +## In[11]:
28 +
29 +class rocket:
30 +    def __init__(self):
31 +        mass = length = radius = F = None
32 +        pos = angle = S = U = V = a = [None, None, None]
33 +
34 +    def get_home_radius(self):
35 +        radius = np.sqrt((self.pos[0]**2)+(self.pos[1]**2)+(self.pos[2]**2))
36 +        return radius
37 +
38 +    def get_grav_vector(self, obj):
39 +        Fgrav = gravity(obj.mass, self.mass, self.get_home_radius)
40 +        Fgravx = Fgrav*(np.sqrt((selfpos[0]**2)/(self.get_home_radius**2)))
41 +        Fgravy = Fgrav*(np.sqrt((selfpos[1]**2)/(self.get_home_radius**2)))
42 +        Fgravz = Fgrav*(np.sqrt((selfpos[2]**2)/(self.get_home_radius**2)))
```

```

43     +         return [Fgravx, Fgravy, Fgravz]
44     +
45     +     def resolve_thrust(self): # yaw, pitch, roll -> x, y, z
46     +         x = self.F*(np.cos(self.angle[0])*np.cos(self.angle[1]))
47     +         y = self.F*(np.sin(self.angle[0]))
48     +         z = self.F*(np.sin(self.angle[0])*np.cos(self.angle[1]))
49     +         return [x, y, z]
50     +
51     +class planet:
52     +    class atmosphere:
53     +        def __init__(self):
54     +
55     +            def atmos_pressure(P0, Mm, T, h): # P0 = Pressure at sea Level
56     +                P = P0 * np.exp((-1 * (Mm * 9.807) / (8.3145 * T)) * h)
57     +                return P
58     +
59     +        def __init__(self, mass, radius, p0, molMass):
60     +            self.mass, self.radius, self.p0, self.molMass = (mass, radius, p0, molMass)
61     +            atmosphere = self.atmosphere()
62     +
63     +
64     +# In[12]:
65     +
66     +objects = []
67     +Earth = planet(5.972e24, 6371e3, 101325, 0.02896)
68     +objects.append(Earth)
69     +
70     +saturnv = rocket()
71     +
72     +
73     +# In[59]:
74     +
75     +import matplotlib.pyplot as plt
76     +from mpl_toolkits.mplot3d import Axes3D
77     +fig = plt.figure()
78     +ax = plt.axes(projection='3d')
79     +
80     +saturnv.F = 500
81     +saturnv.pos = [0, 6371e3, 0]
82     +saturnv.angle = [0, 90, 0]
83     +saturnv.angle = np.radians(saturnv.angle)
84     +x = []
85     +y = []
86     +z = []
87     +for i in range(10):
88     +    F = saturnv.resolve_thrust()
89     +    saturnv.pos[0] = saturnv.pos[0] + F[0]
90     +    print
91     +    saturnv.pos[1] = saturnv.pos[1] + F[1]
92     +    saturnv.pos[2] = saturnv.pos[2] + F[2]
93     +    x.append(saturnv.pos[0])
94     +    y.append(saturnv.pos[1])
95     +    z.append(saturnv.pos[2])
96     +    #saturnv.angle[0] = saturnv.angle[0]+np.radians(0.01)
97     +    #saturnv.angle[1] = saturnv.angle[1]+np.radians(1)
98     +
99     +print(F)
100    +ax.plot(x, y, z, '-b')
101    +ax.set_xlabel('X axis')
102    +ax.set_ylabel('Y axis')
103    +ax.set_zlabel('Z axis')
104    +plt.show()
105    +
106    +
107    +# In[ ]:
108    +
109    +
110    +

```

```
... ... @@ -0,0 +1,8 @@  
1 +  
2 +rocket_x = 0  
3 +rocket_y = 0  
4 +rocket_z = 0  
5 +  
6 +# f = ma  
7 +# a = f/m  
8 +
```

0 comments on commit [e2eab4b](#)

071217E1

Test	Pass/Fail	Description	Evidence
Check that all the equations work together to produce a force value	PASS	The functions should integrate correctly and create a 3D vector force in the positive Y direction to make sure that Force (the main component) can be calculated as designed.	<pre>ngine_test.py [3.0616169978683831e-14, 500.0, 0.0]</pre>
Check a 3D position system can be used and functions correctly with the vector equations	PASS	The position system should record where the rocket is at every iteration. It should clearly show the rocket moving in the +Y direction due to the +Y force. This must be checked to make sure that a position system can be used with the mathematical vector functions	

Tested gravitational orbit with new coordinates

[Browse files](#)

I've written some code to calculate an orbit and graph of the position of the rocket during the orbit to check the gravitational function and the coordinate system work together.

From the graph I believe it does :)

↳ experiments

 meowterspace committed on 8 Dec 2017

1 parent [e2eab4b](#) commit 5e0e96715b55244bd741a349ac4edd9604b27dc6

Showing 7 changed files with 739 additions and 11 deletions.

[Unified](#) [Split](#)

77  Computation Engine/Compute_engine_test.py

```
...
...
@@ -1,4 +1,10 @@
1
2  +# coding: utf-8
3  +
4  +# In[5]:
5  +
6  +
7  +
8  # coding: utf-8
9
10 # In[1]:
11     return F
12
13
14
15
16
17 -def gravity(M, m, r): # P = Pressure, T = temp(k), Vg = flow velocity of gas
18 -    F = (6.67408e-11 * M * m) / (r ** 2) # Cd = Coefficient of drag, A = area in drag
19 -    return F
20 +def gravity(M, m, r):
21 +
22 +    F = (6.67408e-11 * M * m) / (r**2)
23 +    return F
24
25
26
27
28
29 def thrust(q, Ve, Pe, Pa, Ae): # q = rate of ejected mass flow, Ve = exhaust gas ejection speed
30     def __init__(self):
31         mass = length = radius = F = None
32         pos = angle = S = U = V = a = [None, None, None]
33
34
35     def get_home_radius(self):
36         radius = np.sqrt((self.pos[0]**2)+(self.pos[1]**2)+(self.pos[2]**2))
37         return radius
38
39     def get_grav_vector(self, obj):
40         Fgrav = gravity(obj.mass, self.mass, self.get_home_radius)
41         Fgravx = Fgrav*(np.sqrt((self.pos[0]**2)/(self.get_home_radius**2)))
42         Fgravy = Fgrav*(np.sqrt((self.pos[1]**2)/(self.get_home_radius**2)))
43         Fgravz = Fgrav*(np.sqrt((self.pos[2]**2)/(self.get_home_radius**2)))
44
45         Fgrav = gravity(obj.mass, self.mass, self.get_home_radius())
46         #Fgravx = Fgrav*(np.sqrt((self.pos[0]**2)/(self.get_home_radius()**2)))
47         #Fgravy = Fgrav*(np.sqrt((self.pos[1]**2)/(self.get_home_radius()**2)))
48         #Fgravz = Fgrav*(np.sqrt((self.pos[2]**2)/(self.get_home_radius()**2)))
49
50         Fgravx = self.pos[0]/self.get_home_radius()*Fgrav
51         Fgravy = self.pos[1]/self.get_home_radius()*Fgrav
52         Fgravz = self.pos[2]/self.get_home_radius()*Fgrav
53
54         #if Fgravx**2+Fgravy**2+Fgravz**2 != Fgrav**2: print("BAD")
55
56     def resolve_thrust(self): # yaw, pitch, roll -> x, y, z
57
58     class planet:
59
60         class atmosphere:
```

```

53 64         def __init__(self):
54  -         pass
55 66     def atmos_pressure(P0, Mm, T, h): # P0 = Pressure at sea Level
56 67         P = P0 * np.exp((-1 * (Mm * 9.807) / (8.3145 * T)) * h)
57 68         return P
58 77     fig = plt.figure()
59 88     ax = plt.axes(projection='3d')
60 90
61 91     # ---- SIM > Rocket moving through atmosphere
62 92     +"""
63 93     saturnv.F = 500
64 94     saturnv.pos = [0, 6371e3, 0]
65 95     saturnv.angle = [0, 90, 0]
66 96     for i in range(10):
67 97         F = saturnv.resolve_thrust()
68 98         saturnv.pos[0] = saturnv.pos[0] + F[0]
69 100     -     print
70 101     +     print #Something meant to go here?
71 102         saturnv.pos[1] = saturnv.pos[1] + F[1]
72 103         saturnv.pos[2] = saturnv.pos[2] + F[2]
73 104         x.append(saturnv.pos[0])
74 105     ax.set_ylabel('Y axis')
75 106     ax.set_zlabel('Z axis')
76 107     plt.show()
77
78 108     +"""
79
80 109
81 110     # In[ ]:
82 111     # ---- SIM > Orbit (hopefully)
83 112     +saturnv.mass = 500
84 113     +saturnv.a = [0, 0, 0]
85 114     +saturnv.F = 0
86 115     +saturnv.pos = [0, 6779e3, 0] # 408km above Earth's surface
87 116     +saturnv.V = [7800, 0, 0]
88
89 117     +
90 118     +
91 119     +print("{{{{a = "+str(type(saturnv.a).__name__)+"}}}}")
92 120     +print("{{{{+str(type(saturnv.V).__name__)+"}}}}")
93 121     +#print("{{{{+str(type(F).__name__)+"}}}}")
94
95 122     +DATA = []
96 123     +bob = []
97
98 124     +
99 125     +
100 126     +for j in range(100000): #issue[372] increase & see object flyyyyyy into deep space
101 127         F = gravity(Earth.mass, saturnv.mass, saturnv.get_home_radius())
102 128         F = saturnv.get_grav_vector(Earth)
103 129         F = [-F[0],-F[1],-F[2]]
104 130         for i in range(3):
105 131             saturnv.a[i] = (F[i] / saturnv.mass)
106 132             saturnv.V[i] = saturnv.V[i] + saturnv.a[i]*0.1 # 0.1 = tick
107 133             saturnv.pos[i] = saturnv.pos[i] + saturnv.V[i]*0.1
108 134             x = saturnv.pos
109 135             DATA.append([x[0], x[1], x[2]])
110
111 136
112 137
113 138
114 139
115 140
116 141
117 142
118 143
119 144
120 145
121 146
122 147
123 148
124 149
125 150
126 151
127 152
128 153
129 154
130 155
131 156
132 157
133 158
134 159
135 160
136 161

```

	162	+
108	163	
109	164	
110	165	

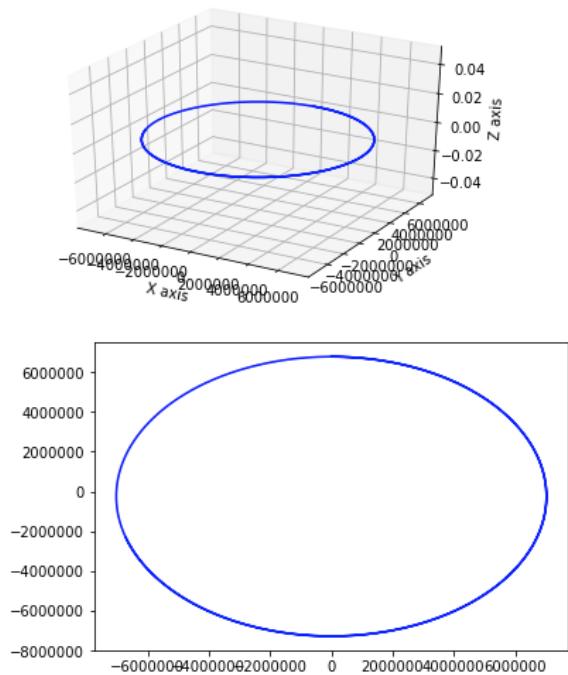
1 comment on commit 5e0e967



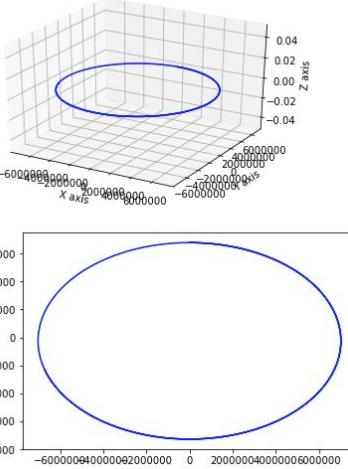
meowterspace commented on 5e0e967 on 8 Dec 2017

Owner

Result:



081217E1

Test	Pass/Fail	Description	Evidence
Check that using the position system a gravitational orbit can be achieved using the gravitational equation	PASS	When plotted it should show a clear elliptical shape existing in only 2 planes. This is to verify that the position system works with the gravitational equations	

261017M2

Append not adding correct data to list #2

[Edit](#)[New issue](#) Closed

meowterspace opened this issue on 8 Dec 2017 · 2 comments



meowterspace commented on 8 Dec 2017

Owner

The code:

```
for j in range(100): #issue[372] increase & see object flyyyyyy into deep space
    F = gravity(Earth.mass, saturnv.mass, saturnv.get_home_radius())
    F = saturnv.get_grav_vector(Earth)
    F = [-F[0],-F[1],-F[2]]
    for i in range(3):
        saturnv.a[i] = (F[i] / saturnv.mass)
        saturnv.V[i] = saturnv.V[i] + saturnv.a[i]*0.1 # 0.1 = tick
        saturnv.pos[i] = saturnv.pos[i] + saturnv.V[i]*0.1
    print(x)
    x = saturnv.pos
    DATA.append(x)
```

returns:

```
[77998.336928211706, 6778562.0077373236, 0.0]
[1559.9999900204971, 6778999.7398036532, 0.0]
[2339.9999600819874, 6778999.4796073064, 0.0]
[3119.9999002049681, 6778999.1326788468, 0.0]
[3899.9998004099361, 6778998.6990182772, 0.0]
[4679.9996507173882, 6778998.1786256013, 0.0]
[5459.9994411478219, 6778997.5715008257, 0.0]
[6239.9991617217356, 6778996.8776439568, 0.0]
[7019.9988024596278, 6778996.097055003, 0.0]
[7799.9983533819986, 6778995.2297339728, 0.0]

[[ 7799.99835338    7799.99835338    7799.99835338    7799.99835338
  7799.99835338    7799.99835338    7799.99835338    7799.99835338
  7799.99835338    7799.99835338]
 [ 6778995.22973397  6778995.22973397  6778995.22973397  6778995.22973397
  6778995.22973397  6778995.22973397  6778995.22973397  6778995.22973397
  6778995.22973397  6778995.22973397]
 [       0.             0.             0.             0.             0.
       0.             0.             0.             0.             0.
     ]]
]]
```

The array of x is clearly changing with every cycle(j) however when appending x to the list DATA, it results in a list of only the final index of x.

I will investigate this further



meowterspace commented on 8 Dec 2017

Owner

```
for i in range(3):
    saturnv.a[i] = (F[i] / saturnv.mass)
    saturnv.V[i] = saturnv.V[i] + saturnv.a[i]*0.1 # 0.1 = tick
    saturnv.pos[i] = saturnv.pos[i] + saturnv.V[i]*0.1
print(DATA)
x = saturnv.pos
DATA.append(x)
```

By printing the array at each iteration of the j for loop, I can see that the list DATA is infact changing but writing over all previous values in the list with the new data?

```
[[1559.9999900204971, 6778999.7398036532, 0.0]]
[[2339.9999600819874, 6778999.4796073064, 0.0], [2339.9999600819874, 6778999.4796073064,
 0.0]]
[[3119.9999002049681, 6778999.1326788468, 0.0], [3119.9999002049681, 6778999.1326788468,
 0.0], [3119.9999002049681, 6778999.1326788468, 0.0]]
[[3899.9998004099361, 6778998.6990182772, 0.0], [3899.9998004099361, 6778998.6990182772,
```

Assignees



No one—assign yourself

Labels



None yet

Projects



None yet

Milestone



No milestone

Notifications

1 participant





meowterspace commented on 8 Dec 2017

Owner

Resolved

Turns out there's something funny about how python's append function works.

The fix was to replace the append line with:

```
DATA.append( [x[0], x[1], x[2]] )
```

The list now contains the correct data :)



 meowterspace closed this on 8 Dec 2017

three.js Navball

[Browse files](#)

I've started development of a Navball (FDAl) to indicate which direction the users rocket is facing. The ball uses three.js (a JS wrapper for WebGL). I chose this over competitors such as Flash because it's very lightweight and has a high compatibility with other browsers. It's composed of a sphere with a custom image mapped onto the surface. The ball rotates in accordance to Euler angles and will update via AJAX.

[experiments](#) meowterspace committed on 30 Dec 2017

1 parent cf8e9ad commit ac8119f40eb1139756ee91157b8ca7544c0745e2

 Showing 5 changed files with 1,034 additions and 0 deletions.[Unified](#) [Split](#)

100 ████████ UI/nav ball/app.js

```
... | ... @@ -0,0 +1,100 @@
1 | +
2 | +
3 | +// -----
4 | +// BASIC SETUP
5 | +// -----
6 | +
7 | +// Create an empty scene
8 | +var scene = new THREE.Scene();
9 | +
10 | +// Create a basic perspective camera
11 | +var camera = new THREE.PerspectiveCamera(50, 400 / 400, 0.1, 1000);
12 | +
13 | +camera.position.z = 10;
14 | +
15 | +// Create a renderer with Antialiasing
16 | +var renderer = new THREE.WebGLRenderer();
17 | +
18 | +// Configure renderer clear color
19 | +//renderer.setClearColor("#000000");
20 | +
21 | +// Configure renderer size
22 | +renderer.setSize( 400, 400 );
23 | +
24 | +// Append Renderer to DOM
25 | +document.body.appendChild( renderer.domElement );
26 | +
27 | +// -----
28 | +// FUN STARTS HERE
29 | +// -----
30 | +
31 | +var cube;
32 | +
33 | +
34 | +// Create a Cube Mesh with basic material 40=50
35 | +var geometry = new THREE.SphereGeometry(3, 32, 32);
36 | +
37 | +// instantiate a loader
38 | +var loader = new THREE.TextureLoader();
39 | +
40 | +// load a resource
41 | +loader.load(
42 | +    // resource URL
43 | +    'map.png',
44 | +
45 | +    // onLoad callback
46 | +    function ( texture ) {
47 | +        // in this example we create the material when the texture is loaded
48 | +        var material = new THREE.MeshBasicMaterial( {
49 | +            map: texture

```

```

50         } );
51
52         cube = new THREE.Mesh( geometry, material );
53
54 // Add cube to Scene
55 +scene.add( cube );
56
57     },
58
59     // onProgress callback currently not supported
60     undefined,
61
62     // onError callback
63     function ( err ) {
64         console.error( 'An error happened.' );
65     }
66
67 );
68
69 +var pgeometry = new THREE.PlaneGeometry(9, 9, 32)
70
71 +loader.load(
72     'shader.png',
73     function (texture) {
74         var pmaterial = new THREE.MeshBasicMaterial({
75             map: texture });
76         plane = new THREE.Mesh( pgeometry, pmaterial );
77 +scene.add( plane );
78 +plane.position.set(0, 0, 0);
79 +plane.transparent = true;
80 });
81
82
83 // Render Loop
84 +var render = function () {
85     requestAnimationFrame( render );
86
87     cube.rotation.x = Math.PI/2;
88     //cube.rotation.y = 90;
89     // Render the scene
90     renderer.render(scene, camera);
91 };
92
93 //var move_to = function() {
94
95
96
97
98 +render();
99
100

```

19 ■■■■■ UI/nav ball/index.html

```

...
...
@@ -0,0 +1,19 @@
1 +<html lang="en">
2 +<head>
3 +  <meta charset="UTF-8" />
4 +  <title>Three.js 101</title>
5 +  <!-- Simple reset to delete the margins -->
6 +  <style>
7 +    body { margin: 0; }
8 +    canvas { width: 100%; height: 100% }
9 +  </style>
10 + <!-- Three.js CDN -->
11 + <script src="three.min.js"></script>
12 +</head>
13 +<body>
14 + <!-- Our code -->
15 + <script src="app.js"></script>

```

```
16 +</body>
17 +</html>
18 +
19 +
```

BIN UI/nav ball/map.png

Binary file not shown.

BIN UI/nav ball/shader.png

Binary file not shown.

915 UI/nav ball/three.min.js

[Load diff](#)

Large diffs are not rendered by default.

3 comments on commit [ac8119f](#)



meowterspace replied on 30 Dec 2017 • edited

Owner

It should be noted that three.js only runs on a web server. This shouldn't be a problem further down the line however for testing purposes a python web server should be set up in the local dir.

```
> cd [dir]
> python3 -m http.server [port] where I suggest using port 8888
```



meowterspace replied on 11 Jan

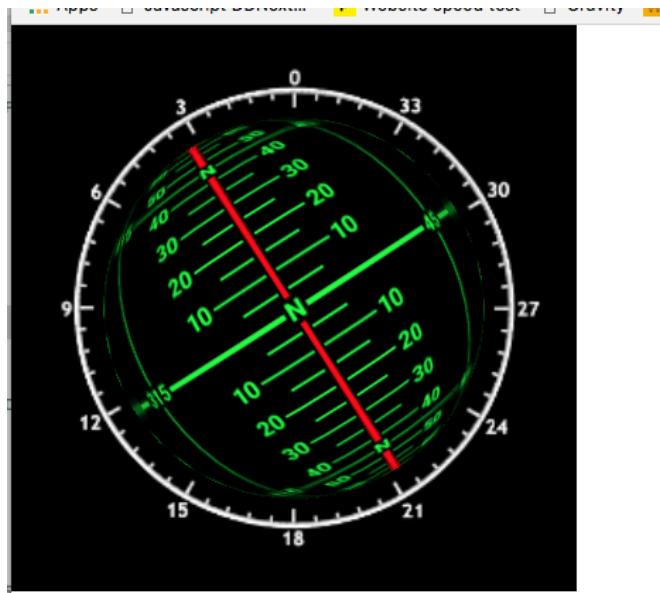
Owner

Preview: https://twitter.com/Meowter_space/status/946141043711410177/video/1

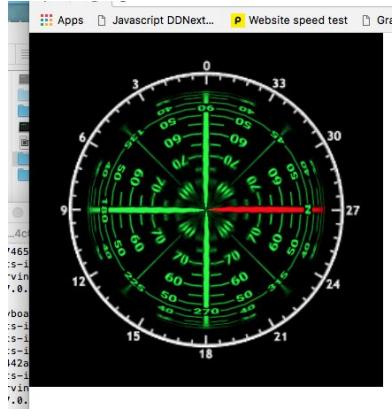
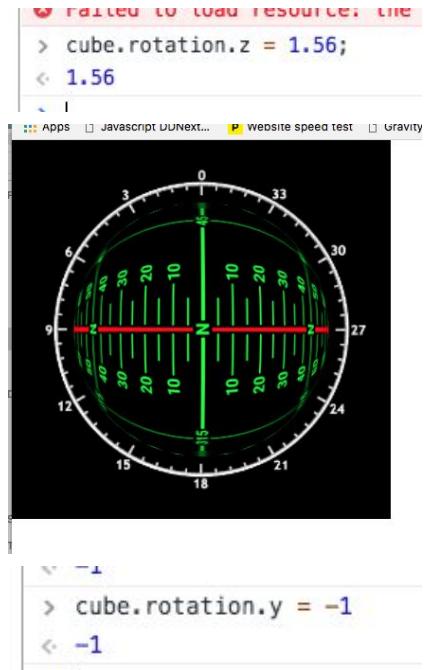
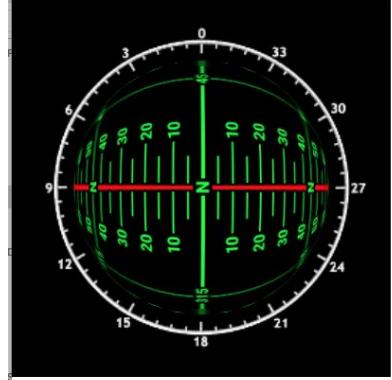


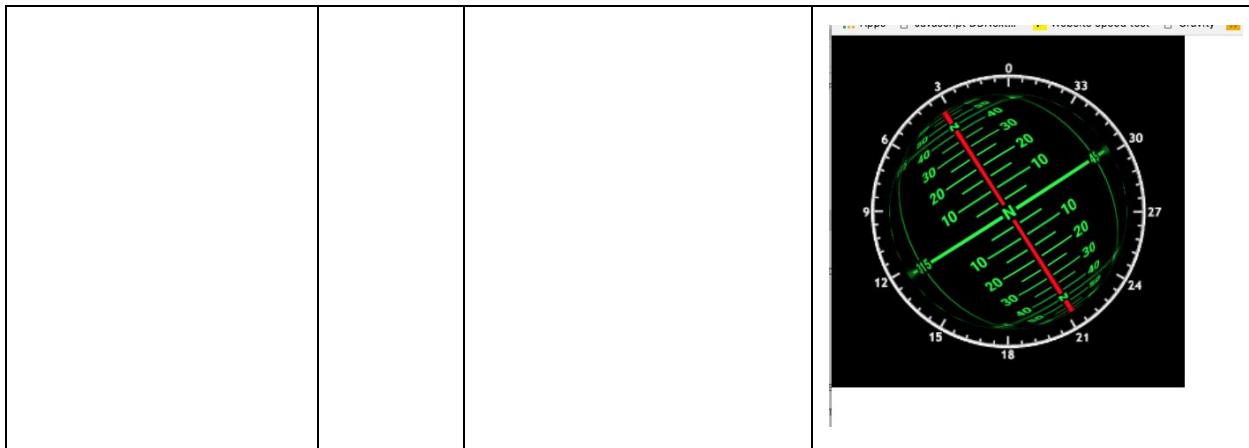
meowterspace replied 19 minutes ago

Owner



301217E1

Test	Pass/Fail	Description	Evidence
Check the page still loads correctly after changes	PASS	The page should load with zero errors to check that the page redirect is correct and accepted by the server	
Check that three.js pages can be loaded on a Flask server	PASS	The page should load without error to check that three.js (and thus WebGL) works on the server. The three.js canvas and navball elements should be visible on the page	
Check that three.js Nav-ball can be rotated using javascript	PASS	The navball should rotate	 <pre>✖ Failed to load resource: the > cube.rotation.z = 1.56; < 1.56 I Apps Javascript DDNext... Website speed test Gravity</pre>  <pre>> cube.rotation.y = -1 < -1 I</pre>



update naval map

[Browse files](#)

1 experiments

 meowterspace committed on 10 Jan

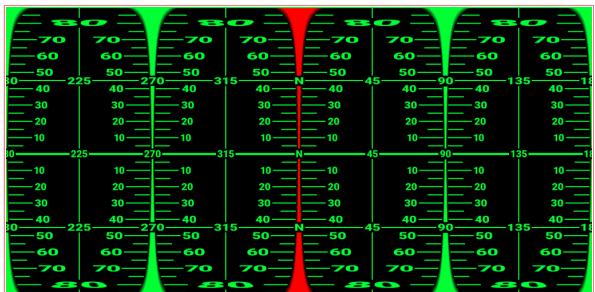
1 parent ac8119f commit b46fc3b4682d3212fa397a756527516dbe222b73

 Showing 1 changed file with 0 additions and 0 deletions.

[Unified](#) [Split](#)

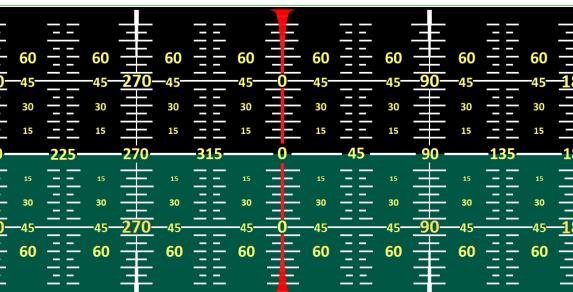
BIN UI/nav_ball/map.png

Deleted



W: 1024px | H: 512px

Added



W: 1024px | H: 512px

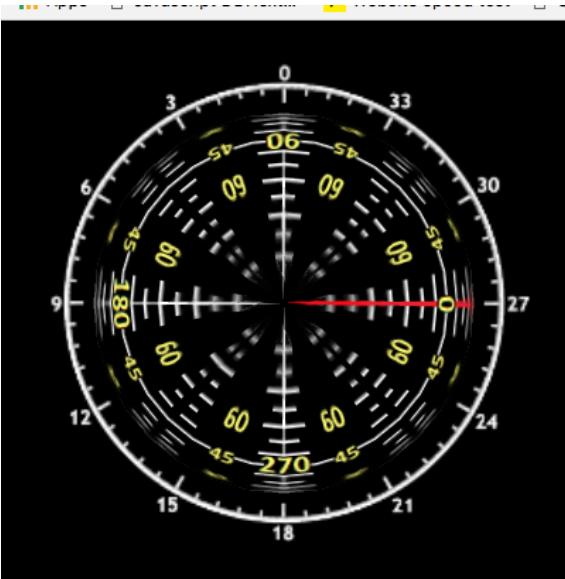
[2-up](#) | [Swipe](#) | [Onion Skin](#)

1 comment on commit b46fc3b



meowterspace commented on b46fc3b 9 minutes ago

Owner



100118E1

Test	Pass/Fail	Description	Evidence
Check that the sphere map can be updated	PASS	The navball should show an updated surface map to check that maps are loaded from the correct directory and that the new map can be used.	

Three.js block not sitting in modal #3

[Edit](#)[New issue](#)[Closed](#)

meowterspace opened this issue on 11 Jan · 4 comments



meowterspace commented on 11 Jan

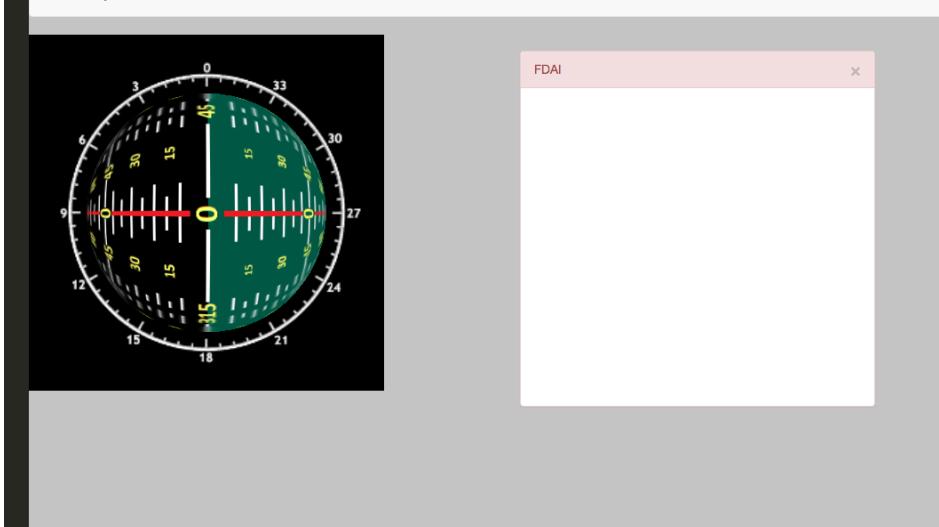
Owner

```

<div class="handle panel-heading">FDI
  <button type="button" class="close" aria-label="Close"
  onclick="hideModal('win-fdai')">
    <span aria-hidden="true">&times;</span>
  </button>
</div>
<div class="panel-body"><script src='resources/js/navball.js'></script>
</div>
</div>

```

From the above code the navball.js script should run inside the panel-body div but instead appears on the main screen, regardless of whether modal is active or not.



I'm not sure why this is but likely has something to do with how three.js renders, and how canvas works.



meowterspace commented on 12 Jan

Owner

Not how javascript works. The code referenced above just calls the script as the panel-body is defined, not as it's rendered. The script should be placed at the top of the HTML file in the head.

Where the three.js rendering is placed is defined in the three.js script at

```
document.body.appendChild( render.domElement );
```

Change this to the modal panel-body, and make the entire three.js navball script a function callable from the modal rendering script in the HTML



meowterspace commented on 12 Jan

Owner

After a lot of debugging the new code works. Variables `renderer` , `scene` , `camera` & `cube` have been declared outside all the functions to act as global variables to accommodate for the scope of the functions.

```

var renderer;
var scene;
var camera;
var cube;

function setup() {

  // Create an empty scene
  scene = new THREE.Scene();

  // Create a basic perspective camera

```

Assignees



No one—assign yourself

Labels



None yet

Projects



None yet

Milestone



No milestone

Notifications

1 participant



```

camera = new THREE.PerspectiveCamera(50, 400 / 400, 0.1, 1000);
camera.position.z = 10;

// Create a renderer with Antialiasing
renderer = new THREE.WebGLRenderer();

// Configure renderer clear color
//renderer.setClearColor("#000000");

// Configure renderer size
renderer.setSize( 400, 400 );

// Append Renderer to DOM
var canvas = document.getElementById('canvas');
canvas.appendChild( renderer.domElement );

};

// -----
// FUN STARTS HERE
// -----
// -----



function draw(){

    // Create a Cube Mesh with basic material 40=50
    var geometry = new THREE.SphereGeometry(3, 32, 32);

    // instantiate a loader
    var loader = new THREE.TextureLoader();

    // load a resource
    loader.load(
        // resource URL
        'resources/img/nav-ball/map.png',

        // onLoad callback
        function ( texture ) {
            // in this example we create the material when the texture is
loaded
            var material = new THREE.MeshBasicMaterial( {
                map: texture
            });

            cube = new THREE.Mesh( geometry, material );

            // Add cube to Scene
            scene.add( cube );

        },
        // onProgress callback currently not supported
        undefined,
        // onError callback
        function ( err ) {
            console.error( 'An error happened.' );
        }
    );

    var pgeometry = new THREE.PlaneGeometry(9, 9, 32)

    loader.load(
        'resources/img/nav-ball/shader.png',
        function (texture) {
            var pmaterial = new THREE.MeshBasicMaterial({
                map: texture });
            plane = new THREE.Mesh( pgeometry, pmaterial );
            scene.add( plane );
            plane.position.set(0, 0, 0);
            plane.transparent = true;
        });
    };

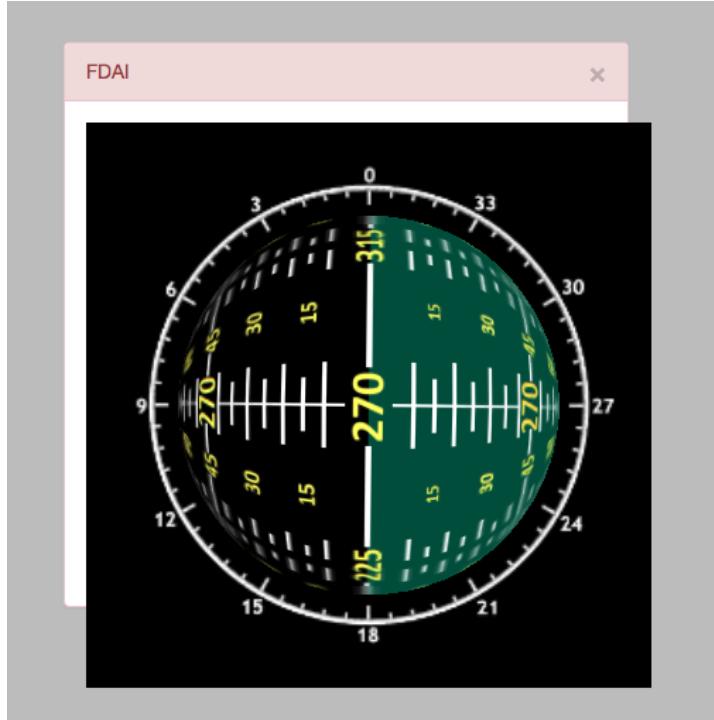
    // Render Loop
    var render = function () {
        requestAnimationFrame( render );

        //cube.rotation.x = Math.PI/2;
        //cube.rotation.y = 90;
        // Render the scene
        renderer.render(scene, camera);
    };
}

```

```
//var move_to = function() {  
  
function render_navball() {  
    setup();  
    draw();  
    render();  
};
```

The navball can now be rendered on demand with the modal, just needs repositioning

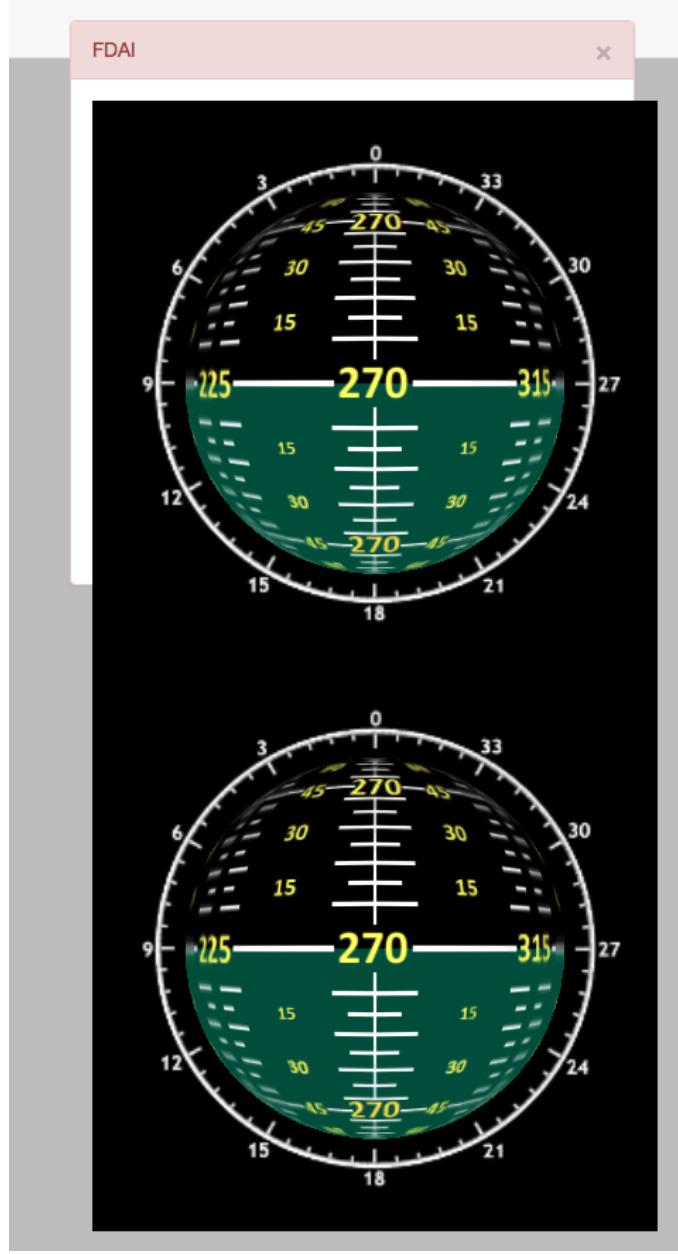


meowterspace commented on 12 Jan

Owner

Bug

Opening, closing and reopening the FDAI causes the three.js canvas to be rendered twice.



meowterspace commented on 12 Jan

Owner

Quick hacky fix:

```
var FDAI_rendered = 0; defined in ui.html
```

```
if (id=='win-fdai' && FDAI_rendered == 0) {  
    render_navball();  
    FDAI_rendered = 1;  
}
```

[^] changed in showModal() to set FDAO_rendered flag to 1 to stop it rendering again



 meowterspace closed this on 12 Jan

Flask not finding /resources folder #4

[Edit](#)[New issue](#)[Closed](#)

meowterspace opened this issue on 12 Jan · 1 comment



meowterspace commented on 12 Jan

Owner

Issue

Flask isn't seeing the /resources folder at '.../client/resources' that contains javascript files and images.

It does however see the HTML file in '.../client' which is set as the `template_folder`.
`app.config['APPLICATION_ROOT'] = '.../client'` seems to make little difference

Config

```
DEBUG True
TESTING False
PROPAGATE_EXCEPTIONS None
PRESERVE_CONTEXT_ON_EXCEPTION None
SECRET_KEY secret
PERMANENT_SESSION_LIFETIME 31 days, 0:00:00
USE_X_SENDFILE False
LOGGER_NAME __main__
LOGGER_HANDLER_POLICY always
SERVER_NAME None
APPLICATION_ROOT ../client
SESSION_COOKIE_NAME session
SESSION_COOKIE_DOMAIN None
SESSION_COOKIE_PATH None
SESSION_COOKIE_HTTPONLY True
SESSION_COOKIE_SECURE False
SESSION_REFRESH_EACH_REQUEST True
MAX_CONTENT_LENGTH None
SEND_FILE_MAX_AGE_DEFAULT 12:00:00
TRAP_BAD_REQUEST_ERRORS False
TRAP_HTTP_EXCEPTIONS False
EXPLAIN_TEMPLATE_LOADING False
PREFERRED_URL_SCHEME http
JSON_AS_ASCII True
JSON_SORT_KEYS True
JSONIFY_PRETTYPRINT_REGULAR True
JSONIFY_MIMETYPE application/json
TEMPLATES_AUTO_RELOAD None
```

Assignees



No one—assign yourself

Labels



None yet

Projects



None yet

Milestone



No milestone

Notifications

1 participant



meowterspace commented on 12 Jan

Owner

Solution

I don't think Flask is allowing indexing of the website (quite sensibly) so i've written some code to do this for me and deliver requested content:

```
import os
import io
from flask import send_file
#-----
@app.route('/')
def index():
    return render_template('ui.html')

@app.route('/resources/img/<path:path>')
def img_route(path):
    ext = os.path.splitext(path[-1].lower())
    if ext == '.jpg' or '.png' or '.gif':
        with open('../client/resources/img/'+str(path), 'rb') as bites:
            return send_file(
                io.BytesIO(bites.read()),
                #attachment_filename=str(path),
                mimetype='image/'+str(ext)
            )
```

```
@app.route('/<path:path>')
def route(path):
    if os.path.exists('../client/'+str(path)) == True:
        return render_template('/'+str(path))
    else:
        return "ERROR 404: "+str(path)+" doesn't exist"
```

 meowterspace closed this on 12 Jan

Bug Fixes and Navball Implementation

[Browse files](#)

I've implemented the three.js navball (see experiments branch), and I've changed the way Flask handles file/path requests as this was causing issues with ui.html finding js files and images, since the server wasn't allowing access to anything other than set as a defined path

by master

 meowterspace committed on 12 Jan

1 parent 7361227 commit 1da36f02cc9270abb4efd4cd2723e8776a459227

Showing 6 changed files with 1,084 additions and 14 deletions.

[Unified](#) [Split](#)

BIN client/resources/img/nav-ball/map.png

Binary file not shown.

BIN client/resources/img/nav-ball/shader.png

Binary file not shown.

111 client/resources/js/navball.js

...	...	@@ -0,0 +1,111 @@
1		+
2		+
3		+// -----
4		+// BASIC SETUP
5		+// -----
6		+
7		+var renderer;
8		+var scene;
9		+var camera;
10		+var cube;
11		+
12		+function setup() {
13		+
14		+ // Create an empty scene
15		+ scene = new THREE.Scene();
16		+
17		+ // Create a basic perspective camera
18		+ camera = new THREE.PerspectiveCamera(50, 400 / 400, 0.1, 1000);
19		+
20		+ camera.position.z = 10;
21		+
22		+ // Create a renderer with Antialiasing
23		+ renderer = new THREE.WebGLRenderer();
24		+
25		+ // Configure renderer clear color
26		+ //renderer.setClearColor("#000000");
27		+
28		+ // Configure renderer size
29		+ renderer.setSize(400, 400);
30		+
31		+ // Append Renderer to DOM
32		+ var canvas = document.getElementById('canvas');
33		+ canvas.appendChild(renderer.domElement);
34		+
35		+};
36		+// -----
37		+// FUN STARTS HERE
38		+// -----
39		+
40		+
41		+function draw(){

```
42 +
43 +
44 +     // Create a Cube Mesh with basic material 40=50
45 +     var geometry = new THREE.SphereGeometry(3, 32, 32);
46 +
47 +     // instantiate a loader
48 +     var loader = new THREE.TextureLoader();
49 +
50 +     // load a resource
51 +     loader.load(
52 +         // resource URL
53 +         'resources/img/nav-ball/map.png',
54 +
55 +         // onLoad callback
56 +         function ( texture ) {
57 +             // in this example we create the material when the texture is loaded
58 +             var material = new THREE.MeshBasicMaterial( {
59 +                 map: texture
60 +             } );
61 +
62 +             cube = new THREE.Mesh( geometry, material );
63 +
64 +             // Add cube to Scene
65 +             scene.add( cube );
66 +
67 +         },
68 +
69 +         // onProgress callback currently not supported
70 +         undefined,
71 +
72 +         // onError callback
73 +         function ( err ) {
74 +             console.error( 'An error happened.' );
75 +         }
76 +
77 +     );
78 +
79 +     var pgeometry = new THREE.PlaneGeometry(9, 9, 32)
80 +
81 +     loader.load(
82 +         'resources/img/nav-ball/shader.png',
83 +         function (texture) {
84 +             var pmaterial = new THREE.MeshBasicMaterial({
85 +                 map: texture });
86 +             plane = new THREE.Mesh( pgeometry, pmaterial );
87 +             scene.add( plane );
88 +             plane.position.set(0, 0, 0);
89 +             plane.transparent = true;
90 +         });
91 +
92 +);
93 +
94 +// Render Loop
95 +var render = function () {
96 +    requestAnimationFrame( render );
97 +
98 +    //cube.rotation.x = Math.PI/2;
99 +    //cube.rotation.y = 90;
100 +    // Render the scene
101 +    renderer.render(scene, camera);
102 +};
103 +
104 +//var move_to = function() {
105 +
106 +function render_navball() {
107 +    setup();
108 +    draw();
109 +    render();
110 +};
111 +
```

915 client/resources/js/three.min.js

Load diff

Large diffs are not rendered by default.

34 client/ui.html

```
11 11          <!-- INCLUDE THE BOOTSTRAP LIBRARIES VIA CDN -->
12 12          <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
13 13          <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
14 14          -
15 15          -
16 14+         <script src="resources/js/three.min.js"></script>
17 15+         <script src='resources/js/navball.js'></script>
18 16          <style type="text/css">
19 17              body {
20 18                  background-color: #c4c4c4;
21 19                  width: 200px;
22 20                  height: 400px;
23 21          }
24 22          #win2 {
25 23              #win-FDAI {
26 24                  width: 400px;
27 25                  height: 200px;
28 26                  height: 400px;
29 27                  margin: 0;
30 28          }
31 29          .panel {
32 30              position: absolute;
33 31              clear: both;
34 32              display: none;
35 33          }
36 34+         #canvas {
37 35+             width: 100%;
38 36+             height: 100%;
39 37+         }
40 38+     }
41 39
42 40         </style>
43 41
44 42         <script>
45 43             var FDAI_rendered = 0;
46 44         </script>
47 45
48 46         <script>
49 47
50 53             var win = document.getElementById(id);
51 54             $(win).css("z-index", ++zIndex);
52 55             $(win).show();
53 56             if (id=='win-fdai' && FDAI_rendered == 0) {
54 57                 render_navball();
55 58                 FDAI_rendered = 1;
56 59             }
57 60         };
58 61
59 62         /* Function for hiding the modal */
60 63
61 64         //document.getElementById('time').innerHTML = 'Time ('+msg.zone+'): '+msg.time;
62 65         document.getElementById('serv').innerHTML = msg.serv;
63 66
64 67         cube.rotation.x = msg.angle[0];
65 68         socket.send(msg.uuid);
66 69         console.log(msg.uuid);
67 70     );
68 71
69 72         <li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown" href="#">
70 73             <ul class="dropdown-menu">
```

```
109 | 123 | <li><a href="javascript:showModal('win1')">T1</a></li>
110 | - | <li><a href="javascript:showModal('win2')">T2</a></li>
111 | 124 | <li><a href="javascript:showModal('win-fdai')">FDAl</a></li>
112 | 125 | <li><a href="#">T3</a></li>
113 | 126 | </ul>
114 | 127 | </li>
115 | 128 | <div class="panel-body">CONTENT</div>
116 | 129 | </div>
117 | 130 | <div id="win2" class="col-2 panel panel-danger">
118 | 131 | <div class="handle panel-heading">Window 2
119 | 132 | <button type="button" class="close" aria-label="Close" onclick="hideModal('win2')">
120 | 133 | <div id="win-fdai" class="col-2 panel panel-danger">
121 | 134 | <div class="handle panel-heading">FDAl
122 | 135 | <button type="button" class="close" aria-label="Close" onclick="hideModal('win-fdai')">
123 | 136 | <span aria-hidden="true">&times;</span>
124 | 137 | </button>
125 | 138 | </div>
126 | 139 | <div class="panel-body">CONTENT</div>
127 | 140 | <div class="panel-body"><div id='canvas'></div></div>
128 | 141 | </div>
129 | 142 | </body>
```

38 server/app.py

```
...  ...
1  1  import json
2  2  import uuid
3  3  import datetime
4  4  -from flask import Flask
5  5  +import os
6  6  +import io
7  7  +from flask import Flask, render_template, send_file
8  8  from flask_socketio import SocketIO, emit, send
9  9  from threading import Thread
10 10
11 11  -app = Flask(__name__)
12 12  +app = Flask(__name__, template_folder='../client')
13 13  app.config['SECRET_KEY'] = 'secret'
14 14  +app.config['DEBUG'] = True
15 15  +#app.config['APPLICATION_ROOT'] = '../client'
16 16  +for i in app.config: print(i, app.config[i])
17 17
18 18  data = {
19 19      'time' : '',      #current Time
20 20      'zone' : 'GMT',  #timezone
21 21      'serv' : '',      #time the server's been active
22 22  -     'uuid' : ''      #unique message id
23 23  +     'uuid' : '',      #unique message id
24 24  +     'angle' : [0,0,0]
25 25
26 26  start = datetime.datetime.now()
27 27  +@app.route('/')
28 28  +def index():
29 29  +    return render_template('ui.html')
30 30
31 31  +@app.route('/resources/img/<path:path>')
32 32  +def img_route(path):
33 33  +    ext = os.path.splitext(path[-1].lower())
34 34  +    if ext == '.jpg' or '.png' or '.gif':
35 35  +        with open('../client/resources/img/'+str(path), 'rb') as bites:
36 36  +            return send_file(
37 37  +                io.BytesIO(bites.read()),
38 38  +                #attachment_filename=str(path),
39 39  +                mimetype='image/'+str(ext)
40 40  +            )
```

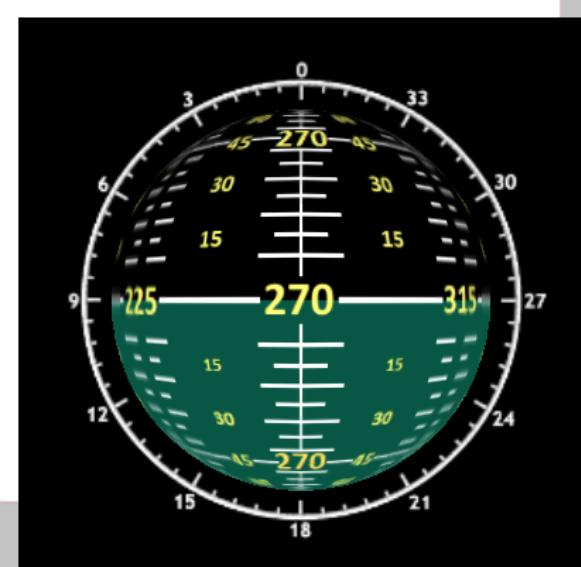
```
41  +
42  +@app.route('/<path:path>')
43  +def route(path):
44  +  if os.path.exists('../client/'+str(path)) == True:
45  +    return render_template('/'+str(path))
46  +  else:
47  +    return "ERROR 404: "+str(path)+" doesn't exist"
48  +
20
49
50  @socketio.on('message')
51  def handle_message(message):
52    data['time'] = str(datetime.datetime.now())
53    data['serv'] = str(datetime.datetime.now()-start)
54    data['uuid'] = str(uuid.uuid4())
55    data['angle'][0] = data['angle'][0]+0.01
56    send(data)
57
58
59
60
61
62
63  if __name__ == '__main__':
64    - socketio.run(app) ❷
65    + socketio.run(app)
```

1 comment on commit 1da36f0



meowterspace commented on 1da36f0 14 minutes ago

Owner

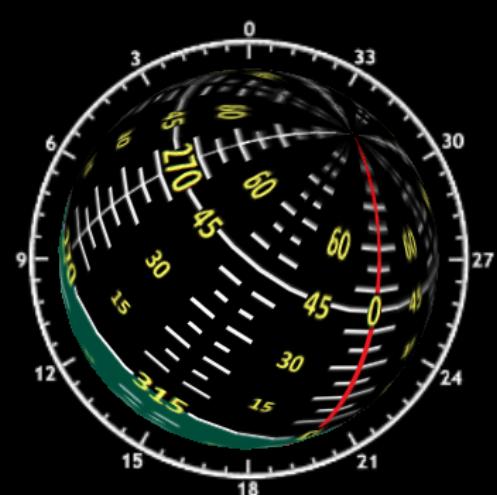


• [y11u11A1](#)

T1

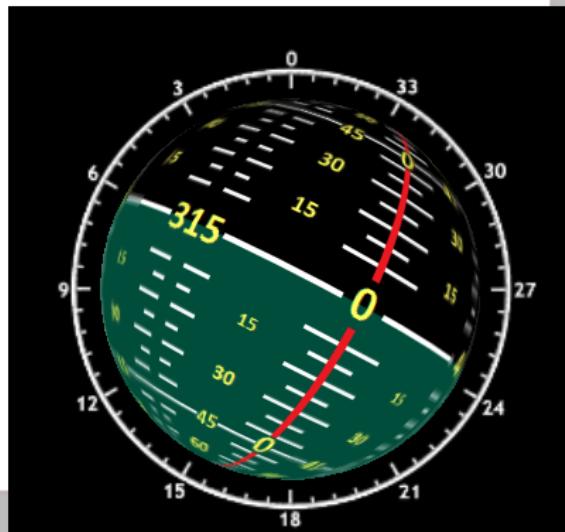
FDAI

T3



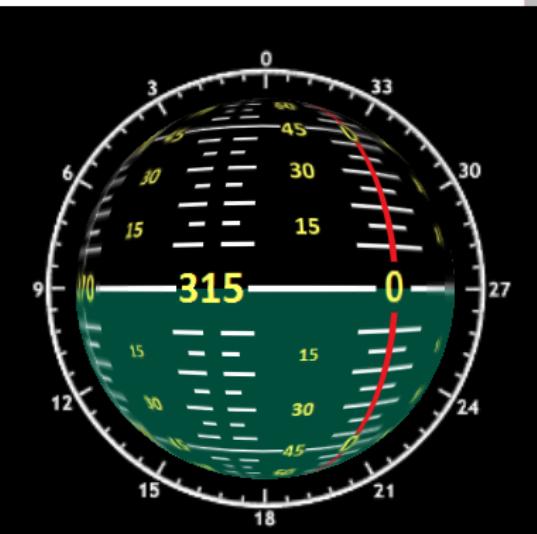
FDAI

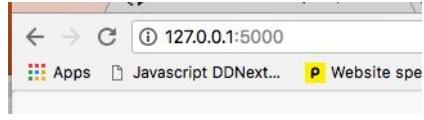
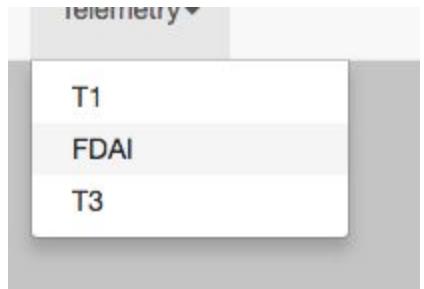
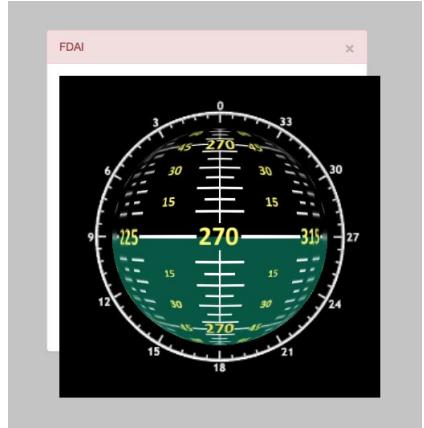
X

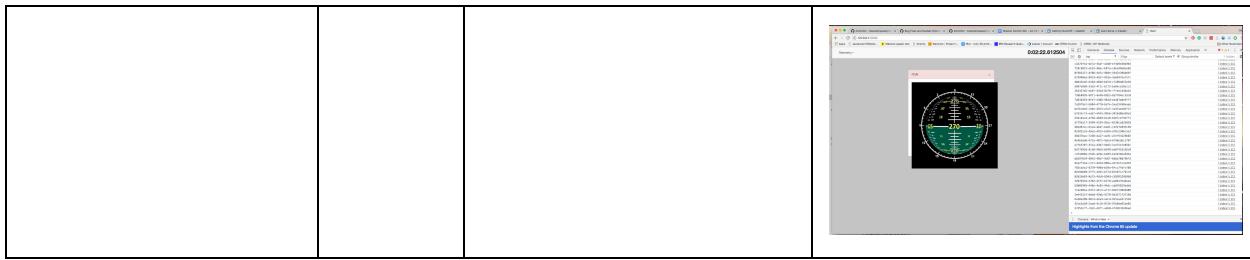


FDAI

X



Test	Pass/Fail	Description	Evidence
Check the page loads correctly after changes	PASS	The page should load with zero errors to check that the page redirect is correct and accepted by the server	
In /ui.html, check a nav-ball button can be added to navigation bar	PASS	A nav-ball button should appear in a drop down menu of the navbar to make sure the UI functions with new changes	
In /ui.html, check a three.js nav-ball canvas can be integrated into a modal	PASS	The three.js navball and canvas should appear within the modal and can be dragged with the modal. This is an important usability feature.	
In /ui.html, check that nav-ball can rotate according to defined server data over websockets	PASS	The navball should spin in accordance to websocket data. This ensures that websocket data can be used to influence and change three.js entities.	<pre>f7440056e-0f79-492c-a829-4cc6881a433c1 753970f8-4248-4390-a21c-1f86f0a7f234d e035902e-fee0-4739-8599-4e122092404d e91ac2ab-caff-46c7-bc48-4d75d39762a1 cf108d48-3eaf-41b6-815c-219767d398e6 e035902e-fee0-4739-8599-4e122092404d 82299738-089d-48ed-b7c0-33d9baad1d4 a436c3c7-779d-492c-8a40-9919fb313a0d 53f0cdd6-0873-4c97-8f72-c23c88fb7e6 36111a6d-6346-4922-b2af-194556310778 40761eb0-5d3d-41ac-8259-7f499583c2e8 40761eb0-5d3d-41ac-8259-7f499583c2e8 fd949f57-a3f8-493b-95c1-d169f97f76376 53448f3f-7e34-4c5c-aec9-2129c989fcf6 53448f3f-7e34-4c5c-aec9-2129c989fcf6 39c56933e-aec1-4ef1-bb10-c44fb0bd24a 7d18e082-d05f-4056-8e91-dc62928c2b55 7d18e082-d05f-4056-8e91-dc62928c2b55 b05138b7-a332-439b-b0f1-528fdeca8f52 3e721a02-3627-441b-8e0c-cf5b895e679 243d98e0-451c-4a61-a699-662a3c3a51521 243d98e0-451c-4a61-a699-662a3c3a51521 67449077-cd5f-4fb4-880c-521dd05959391 b0546f11-2a2d-45b5-a25a-4616c93441e8 b0546f11-2a2d-45b5-a25a-4616c93441e8 93f0699c-c53c-48b0-acdf-f784c0d87be2 874e2108-62c6-4e64-9175-7755cfab6eb8 874e2108-62c6-4e64-9175-7755cfab6eb8 d255377d-64d5-4ef4-9513-56c7732e2c1c 91f136e8-d2c2-4c7a-a4f7-c5849886dc3 91f136e8-d2c2-4c7a-a4f7-c5849886dc3 bccc0956-67a1-496d-b27d-9141a861e907 7c278258-6c1b-456c-8976-1d9a5e7e1468 7c278258-6c1b-456c-8976-1d9a5e7e1468 127.0.0.1 - - [11/Apr/2018 14:41:41] "GET /socket.io/?EIO=3&transport=websocket&sid=361371 84e198123cb120a35c8a 200 0 28.260986 "</pre>



Added Lobby

[Browse files](#)

I've added a lobby.html page. This communicates currently connected players and game settings to the page for the user to view via web sockets to give real time feedback of who is connected. I've used a different namespace (channel) for the web socket communications called 'lobbu' to make it easier to differentiate between this and other game web socket namespaces.

I've also now included index.html which will be the first page the user sees when they connect. I have added the required menu and username input field, which is sent to the server by POST and set as a session variable. This will allow me to keep track of which user is which. I have also added a function to instantiate a user object at this point in the name of the user, which i will use later. I have added this because i need to keep track of all my users settings separately.

Finally I have added a CONFIG.txt file to the server folder. This will allow an admin to change essential server settings with ease before the server is started. I have put it in a text file because it is easier and safer to edit than the code for new users, and will be in a familiar format.

by master

 meowterspace committed on 28 Jan

1 parent 1da36f0 commit c0a71eec935015f3ebf1adb4830b7bdf6653d28a

Showing 5 changed files with 404 additions and 3 deletions.

[Unified](#) [Split](#)

163  client/index.html

[Load diff](#)

Large diffs are not rendered by default.

52  client/lobby.html

...	...	@@ -0,0 +1,52 @@
1	1	+<head>
2	2	+<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
3	3	+<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
4	4	+<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/1.4.8/socket.io.min.js"></script>
5	5	+</head>
6	6	+<body>
7	7	+<script type="text/javascript">
8	8	+\$(<document>).ready(function() {
9	9	+ var lobby = io.connect('http://127.0.0.1:5000/lobbu');
10	10	+
11	11	+
12	12	+ lobby.on('connect', function() {
13	13	+ lobby.send('hi');
14	14	+
15	15	+
16	16	+ });
17	17	+
18	18	+ lobby.on('message', function(msg) {
19	19	+ if (String(msg.ACTIVE) == 'true') {
20	20	+ window.location.replace("http://127.0.0.1:5000/ui.html");
21	21	};
22	22	+
23	23	+ var user_table = '<table style="border: 1px solid #000000">';
24	24	+ for (var i=1; i <= msg.USER_LIST.length; i++) {
25	25	+ user_table = user_table + '<tr><td>' + msg.USER_LIST[i-1] + '</td></tr>';
26	26	+ };
27	27	+ user_table = user_table + '</table>';
28	28	+ document.getElementById('party').innerHTML = user_table;

```

29 +     lobby.send('hi');
30 +
31 +     var settings_table = `<table style="border: 1px solid #000000">
32 +       <tr><td>Game Name</td><td>`+String(msg.GAME_NAME)+`</td></tr>
33 +       <tr><td>Description</td><td>`+String(msg.DESCRIPTION)+`</td></tr>
34 +       <tr><td>Difficulty</td><td>`+String(msg.DIFFICULTY)+`</td></tr>
35 +       <tr><td>Online mode</td><td>`+String(msg.ONLINE_MODE)+`</td></tr>
36 +       <tr><td>White list</td><td>`+String(msg.WHITE_LIST)+`</td></tr>
37 +       <tr><td>Cheats</td><td>`+String(msg.CHEATS)+`</td></tr>
38 +       <tr><td>Max game length</td><td>`+String(msg.MAX_GAME_LENGTH)+`</td></tr>
39 +       <tr><td>Scinario</td><td>`+String(msg.SCINARIO)+`</td></tr>
40 +       <tr><td>Active</td><td>`+String(msg.ACTIVE)+`</td></tr>
41 +
42 +     `;
43 +
44 +
45 +     document.getElementById('settings').innerHTML = settings_table;
46 +   });
47 +});
48 +</script>
49 +<div style="float:left" id='settings'></div>
50 +<div style="float:left; padding-left:10px" id='party'></div>
51 +
52 +</body> ④

```

24 client/resources/js/anykey.js

```

... ... @@ -0,0 +1,24 @@
1 +var menu_flag = 0;
2 +
3 +$(document).on('keyup',function(evt) {
4 +  if (menu_flag == 0){
5 +    if (evt.keyCode == 8||evt.keyCode == 9||evt.keyCode == 13||evt.keyCode == 16||evt.keyCode == 17||evt.keyCode ==
6 +
7 +    $('#menu').show();
8 +    $('#PAKTC').hide();
9 +    menu_flag = 1;
10 +  };
11 +};
12 +});
13 +
14 +
15 +function join() {
16 +  $('#menu').hide();
17 +  $('#join-menu').show();
18 +};
19 +function host() {
20 +  alert('host');
21 +};
22 +function about() {
23 +  alert('host');
24 +}; ④

```

28 server/CONFIG.txt

```

... ... @@ -0,0 +1,28 @@
1 +JSON_AS_ASCII = True
2 +USE_X_SENDFILE = False
3 +SESSION_COOKIE_PATH = None
4 +SESSION_COOKIE_DOMAIN = None
5 +SESSION_COOKIE_NAME = session
6 +DEBUG = False
7 +LOGGER_HANDLER_POLICY = always
8 +LOGGER_NAME = None
9 +SESSION_COOKIE_SECURE = False
10 +SECRET_KEY = None
11 +EXPLAIN_TEMPLATE_LOADING = False
12 +MAX_CONTENT_LENGTH = None

```

```
13 +PROPAGATE_EXCEPTIONS = None
14 +APPLICATION_ROOT = None
15 +SERVER_NAME = None
16 +PREFERRED_URL_SCHEME = http
17 +JSONIFY_PRETTYPRINT_REGULAR = True
18 +TESTING = False
19 +PERMANENT_SESSION_LIFETIME = 31 days, 0:00:00
20 +TEMPLATES_AUTO_RELOAD = None
21 +TRAP_BAD_REQUEST_ERRORS = False
22 +JSON_SORT_KEYS = True
23 +JSONIFY_MIMETYPE = application/json
24 +SESSION_COOKIE_HTTPONLY = True
25 +SEND_FILE_MAX_AGE_DEFAULT = 12:00:00
26 +PRESERVE_CONTEXT_ON_EXCEPTION = None
27 +SESSION_REFRESH_EACH_REQUEST = True
28 +TRAP_HTTP_EXCEPTIONS = False
```

140  server/app.py

```
3 3 import datetime
4 4 import os
5 5 import io
6 -from flask import Flask, render_template, send_file
7 +import errno
8 +from flask import Flask, render_template, send_file, request, session, redirect, render_template_string
9 9 from flask_socketio import SocketIO, emit, send
10
11 +header = ''
12 +<head>
13 +<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
14 +<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
15 +<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/1.4.8/socket.io.min.js"></script>
16 +</head>
17 +<body>
18 +<script type="text/javascript">
19 +$(document).ready(function() {
20 +  var lobby = io.connect('http://127.0.0.1:5000/lobbu');
21 +
22 +
23 +  lobby.on('connect', function() {
24 +    lobby.send('hi');
25 +
26 +
27 +  });
28 +
29 +  lobby.on('message', function(msg) {
30 +    var user_table = '<table style="border: 1px solid #000000">';
31 +    for (var i=1; i <= msg.USER_LIST.length; i++) {
32 +      user_table = user_table + '<tr><td>' + msg.USER_LIST[i-1] + '</td></tr>';
33 +    }
34 +    user_table = user_table + '</table>';
35 +    document.getElementById('party').innerHTML = user_table;
36 +    lobby.send('hi');
37 +  });
38 +});
39 +</script>
40 +<div id='party'></div>
41 +</body>
42 +
43 +'''
44 +
45 +CONFIG = {
46 +  'JSON_AS_ASCII': True,
47 +  'USE_X_SENDFILE': False,
48 +  'SESSION_COOKIE_PATH': None,
49 +  'SESSION_COOKIE_DOMAIN': None,
50 +  'SESSION_COOKIE_NAME': 'session',
51 +  'DEBUG': False,
52 +  'LOGGER_HANDLER_POLICY': 'always',
```

```

53     + 'LOGGER_NAME': None,
54     + 'SESSION_COOKIE_SECURE': False,
55     + 'SECRET_KEY': None,
56     + 'EXPLAIN_TEMPLATE_LOADING': False,
57     + 'MAX_CONTENT_LENGTH': None,
58     + 'PROPAGATE_EXCEPTIONS': None,
59     + 'APPLICATION_ROOT': None,
60     + 'SERVER_NAME': None,
61     + 'PREFERRED_URL_SCHEME': 'http',
62     + 'JSONIFY_PRETTYPRINT_REGULAR': True,
63     + 'TESTING': False,
64     + 'PERMANENT_SESSION_LIFETIME': datetime.timedelta(31),
65     + 'TEMPLATES_AUTO_RELOAD': None,
66     + 'TRAP_BAD_REQUEST_ERRORS': False,
67     + 'JSON_SORT_KEYS': True,
68     + 'JSONIFY_MIMETYPE': 'application/json',
69     + 'SESSION_COOKIE_HTTPONLY': True,
70     + 'SEND_FILE_MAX_AGE_DEFAULT': datetime.timedelta(0, 43200),
71     + 'PRESERVE_CONTEXT_ON_EXCEPTION': None,
72     + 'SESSION_REFRESH_EACH_REQUEST': True,
73     + 'TRAP_HTTP_EXCEPTIONS': False
74 }
75 +
76 +GAME = {
77     + 'GAME_NAME': None,
78     + 'DESCRIPTION': None,
79     + 'DIFFICULTY': 0,
80     + 'ONLINE_MODE': True,
81     + 'WHITE_LIST': False,
82     + 'BLACK_LIST': False,
83     + 'CHEATS': False,
84     + 'MAX_GAME_LENGTH': None,
85     + 'BANNED_USERNAMES': False,
86     + 'SCENARIO': False,
87     + 'ACTIVE': False,
88     + 'USER_LIST': []
89 }
90 +
91 +users = []
92 +
93 +class User:
94     + def __init__(self, name):
95         + self.name = name
96     +
97     +
98     +# MAKE CONFIG FILE
99     +if os.path.exists('CONFIG.txt') == True:
100     + with open('CONFIG.txt', 'r') as f:
101         + for line in f.readlines():
102             + line_split = line.strip().split("=")
103             + CONFIG[line_split[0].strip()] = line_split[1].strip()
104         + f.close()
105     +else:
106     + with open('CONFIG.txt', 'w') as f:
107         + for key in CONFIG:
108             + f.write(key+' = '+str(CONFIG[key])+'\n')
109         + f.close()
110     +
111     app = Flask(__name__, template_folder='..client')
112     app.config['SECRET_KEY'] = 'secret'
113     -app.config['DEBUG'] = True
114     +
115     +app.config['DEBUG'] = CONFIG['DEBUG']
116     +
117     #app.config['APPLICATION_ROOT'] = '..client'
118     +for i in app.config: print(i, app.config[i])
119     +
120     socketio = SocketIO(app)
121     +

```

```

18 122 data = {
19 123     'time' : '',      #current Time
20 124     'zone' : 'GMT', #timezone
21 128 }
22 129
23 130     start = datetime.datetime.now()
24 131 +
25 132 +===== APP ROUTES - FLASK =====
26 133 +
27 134     @app.route('/')
28 135     def index():
29 136         - return render_template('ui.html')
30 137         + return render_template('index.html')
31 138 +
32 139     +@app.route('/login', methods=['POST'])
33 140     +def login():
34 141         + if request.method == "POST":
35 142             + session['username'] = request.form['username']
36 143             + username = session['username']
37 144             + exec(str(username)+"=User('"+str(username)+"')")
38 145             +
39 146             + print(session['username'])
40 147             + exec("print(\""+str(username)+".name\")")
41 148             +
42 149             + GAME['USER_LIST'].append(username)
43 150             + print(session)
44 151             + return redirect('/lobby')
45 152             + return redirect(url_for('/'))
46 153             +
47 154     +@app.route('/lobby')
48 155     +def lobby():
49 156         + if session['username'] != None:
50 157             + #return render_template_string(header)
51 158             + return render_template('lobby.html')
52 159
53 160     @app.route('/resources/img/<path:path>')
54 161     def img_route(path):
55 162         + else:
56 163             return "ERROR 404: "+str(path)+" doesn't exist"
57 164
58 165 +===== APP ROUTES - SOCKETIO =====
59 166
60 167     @socketio.on('message')
61 168     def handle_message(message):
62 169         send(data)
63 170
64 171     +@socketio.on('message', namespace='/lobbu')
65 172     +def handle_lobby_message(message):
66 173         + send(GAME)
67 174         +
68 175         + if __name__ == '__main__':
69 176             socketio.run(app)
70 177

```

1 comment on commit [c0a71ee](#)



meowterspace commented on c0a71ee 23 minutes ago

Owner

Added Lobby - meowter 127.0.0.1:5000/lobby

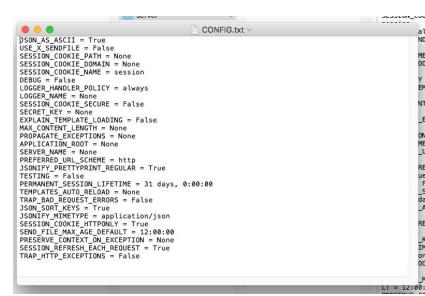
Added Lobby - meowter 127.0.0.1:5000/lobby

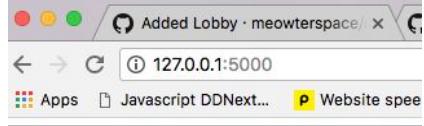
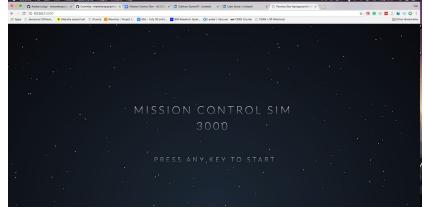
Apps Javascript DDNext... P Apps Javascript DDNext... P Webs

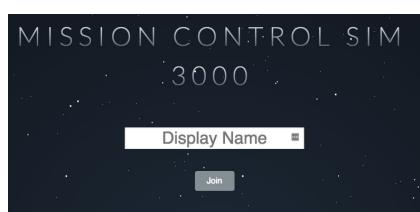
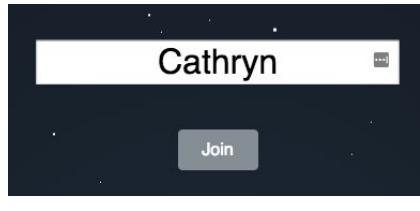
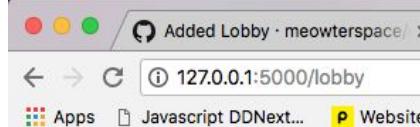
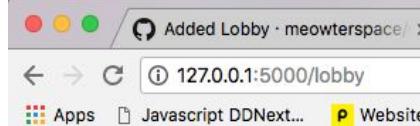
Game Name	null	Cathryn
Description	null	bob
Difficulty	0	
Online mode	true	
White list	false	
Cheats	false	
Max game length	null	
Scinario	false	
Active	false	

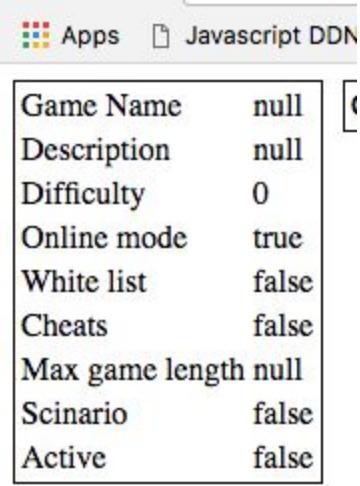
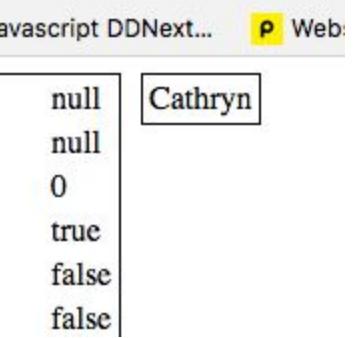
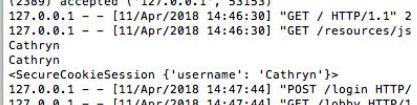
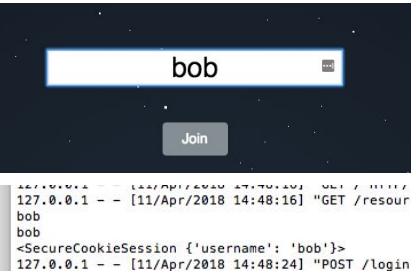
Game Name	null	Cathryn
Description	null	bob
Difficulty	0	
Online mode	true	
White list	false	
Cheats	false	
Max game length	null	
Scinario	false	
Active	false	

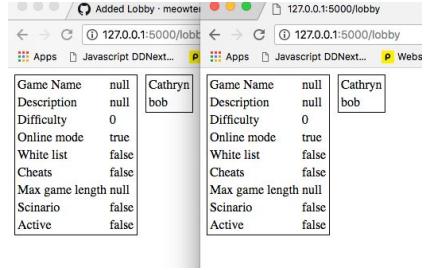
280118M1

Test	Pass/Fail	Description	Evidence
Check that a flask server can be started	PASS	Look at the server terminal. The server should quote a debugger pin and a server address. Important to check the server works before continuing.	<pre> * Debugger is active! * Debugger PIN: 132-511-971 (2389) wsgi starting up on http://127.0.0.1:5000</pre>
Check that the server can parse text and load settings from CONFIG.txt into a dictionary	PASS	Check the server terminal readout. The printed settings should match those identically in the CONFIG.txt file. This test makes sure that settings can be loaded correctly.	

			<pre>Cats-iMac:server Cat\$ python3 app.py DEBUG False TESTING False PROPAGATE_EXCEPTIONS None PRESERVE_CONTEXT_ON_EXCEPTION None SECRET_KEY secret PERMANENT_SESSION_LIFETIME 31 days, 0:00:00 USE_X_SENDFILE False LOGGER_NAME __main__ LOGGER_HANDLER_POLICY always SERVER_NAME None APPLICATION_ROOT None SESSION_COOKIE_NAME session SESSION_COOKIE_DOMAIN None SESSION_COOKIE_PATH None SESSION_COOKIE_HTTPONLY True SESSION_COOKIE_SECURE False SESSION_REFRESH_EACH_REQUEST True MAX_CONTENT_LENGTH None SEND_FILE_MAX_AGE_DEFAULT 12:00:00 TRAP_BAD_REQUEST_ERRORS False TRAP_HTTP_EXCEPTIONS False EXPLAIN_TEMPLATE_LOADING False PREFERRED_URL_SCHEME http JSON_AS_ASCII True JSON_SORT_KEYS True JSONIFY_PRETTYPRINT_REGULAR True JSONIFY_MIMETYPE application/json TEMPLATES_AUTO_RELOAD None * Restarting with stat DEBUG False TESTING False PROPAGATE_EXCEPTIONS None PRESERVE_CONTEXT_ON_EXCEPTION None SECRET_KEY secret PERMANENT_SESSION_LIFETIME 31 days, 0:00:00 USE_X_SENDFILE False LOGGER_NAME __main__ LOGGER_HANDLER_POLICY always SERVER_NAME None APPLICATION_ROOT None SESSION_COOKIE_NAME session SESSION_COOKIE_DOMAIN None SESSION_COOKIE_PATH None SESSION_COOKIE_HTTPONLY True SESSION_COOKIE_SECURE False SESSION_REFRESH_EACH_REQUEST True MAX_CONTENT_LENGTH None SEND_FILE_MAX_AGE_DEFAULT 12:00:00 TRAP_BAD_REQUEST_ERRORS False TRAP_HTTP_EXCEPTIONS False EXPLAIN_TEMPLATE_LOADING False PREFERRED_URL_SCHEME http JSON_AS_ASCII True JSON_SORT_KEYS True JSONIFY_PRETTYPRINT_REGULAR True JSONIFY_MIMETYPE application/json TEMPLATES_AUTO_RELOAD None * Debugger is active! * Debugger PIN: 132-511-971 (2389) wsgi starting up on http://127.0.0.1:5000</pre>
Check that an index page loads	PASS	The page should load without error to make sure that rendering index.html causes no server issues.	
Check that the index page renders the defined index page	PASS	The correct index.html page should be rendered at / to make sure the flask app routes are working correctly.	

Check that the index page has a functioning press any key to continue feature	PASS	Each key should trigger the Join/Host/About menu to appear. It's important to check everykey to make sure the function works across the board.	
Check join button functions	PASS	A text box should take its place. The join button is needed to function to select what mode you want.	
Check name text-field works	PASS	Every character should work fine. This is to make sure the game works in different countries with different keyboards.	
Check the final join button redirects to lobby	PASS	The page should redirect to /lobby. This test is needed to verify that the page routing works correctly	
Check /lobby loads correctly	PASS	The page should load with zero errors to check that the page redirect is correct and accepted by the server	

Check lobby displays game settings	PASS	The lobby should show the current game settings to help the players understand how their specific game will work so they can develop a strategy going in	
Check lobby displays currently connected users	PASS	The lobby should show a table of all the currently connected users, to let the players know when to start the game and who they're playing with	
Check that the server recognises a user has connected	PASS	Observe server terminal readout and the name of the connected user should be printed when they connect. This checks that the /lobbu namespace is working and messages can be transmitted on a seperate websocket channel	 12345 accepted ('127.0.0.1', 55153) 127.0.0.1 - - [11/Apr/2018 14:46:30] "GET / HTTP/1.1" 2 127.0.0.1 - - [11/Apr/2018 14:46:30] "GET /resources/js/Cathryn/Cathryn <SecureCookieSession {'username': 'Cathryn'}> 127.0.0.1 - - [11/Apr/2018 14:47:44] "POST /login HTTP/ 127.0.0.1 - - [11/Apr/2018 14:47:44] "GET /lobby HTTP/1
Check the server allows multiple users to be added	PASS	Each username should be printed in the server terminal in the order they join. This makes sure that multiple users can join	 12345 accepted ('127.0.0.1', 55153) 127.0.0.1 - - [11/Apr/2018 14:48:16] "GET /resource/bob/bob <SecureCookieSession {'username': 'bob'}> 127.0.0.1 - - [11/Apr/2018 14:48:24] "POST /login

All clients update data instantly	PASS	Observe both old and new tabs and both username lists should update simultaneously in real time.	
-----------------------------------	-------------	--	---

Added orbit check

[Browse files](#)

To add an objective to my game, I've added a new orbit check function which checks if a rocket is in a suitable orbit (where the apoapsis is greater than the radius of the edge of the planet's atmosphere).

[experiments](#) meowterspace committed on 15 Mar

1 parent b46fc3b commit 2073659d5026de27364de6460cc1bab436dc38f2

 Showing 1 changed file with 14 additions and 0 deletions.[Unified](#) [Split](#)14  Computation Engine/Compute_engine_test.py

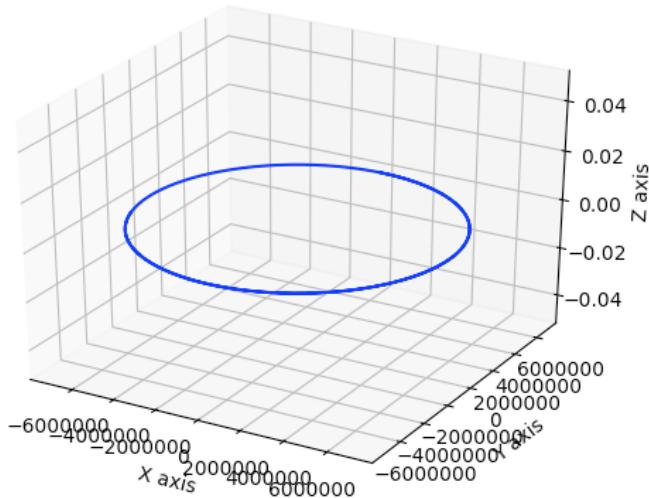
```
33 33     F = (6.67408e-11 * M * m) / (r**2)
34 34     return F
35 35
36 36     +def is_orbit(v, r, m, h):
37 37     +     y = (v[0]*v[1]+v[1]*r[1]+v[2]+r[2])
38 38     +     y = y / (np.sqrt((v[0]**2)+(v[1]**2)+(v[2]**2))+np.sqrt((r[0]**2)+(r[1]**2)+(r[2]**2)))
39 39
40 40     +     v = np.sqrt((v[0]**2)+(v[1]**2)+(v[2]**2))
41 41     +     r = np.sqrt((r[0]**2)+(r[1]**2)+(r[2]**2))
42 42
43 43     +     p = ((6.67408e-11*m - np.sqrt((6.67408e-11**2)*(m**2)-(v**2)*(r**2)*(((2*6.67408e-11*m)/r)-(v**2)))*(1-(np.cos(y))**2))
44 44     +     print(p)
45 45     +     if (p > h): return True
46 46     +     else: return False
47
48 48     def thrust(q, Ve, Pe, Pa, Ae): # q = rate of ejected mass flow, Ve = exhaust gas ejection speed
49 49         F = q * Ve + (Pe - Pa) * Ae # Pe = pressure of exhaust gasses, Pa = pressure of ambient atmosphere
171 182
172 183
173 184     print(distance(saturnv, Earth))
185     +print('__orbit__')
186     +print(is_orbit(saturnv.V, saturnv.pos, 5.972e24, 100000))
187
174 188
```

1 comment on commit 2073659

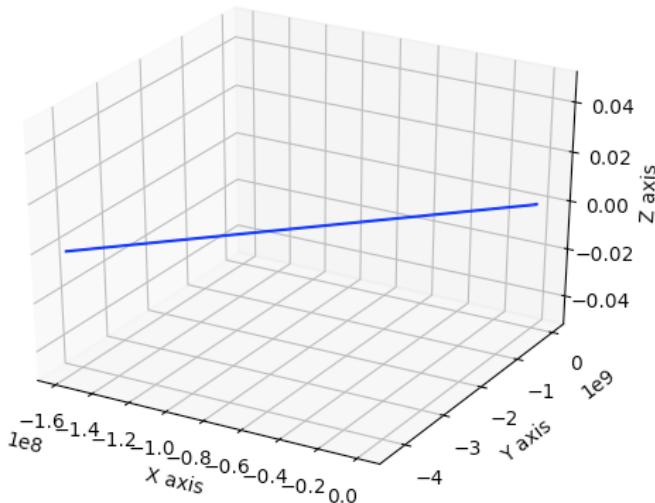


meowterspace commented on 2073659 11 minutes ago

Owner

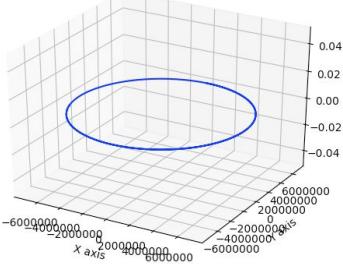
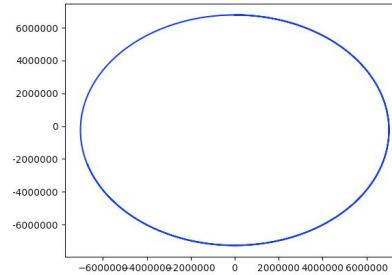


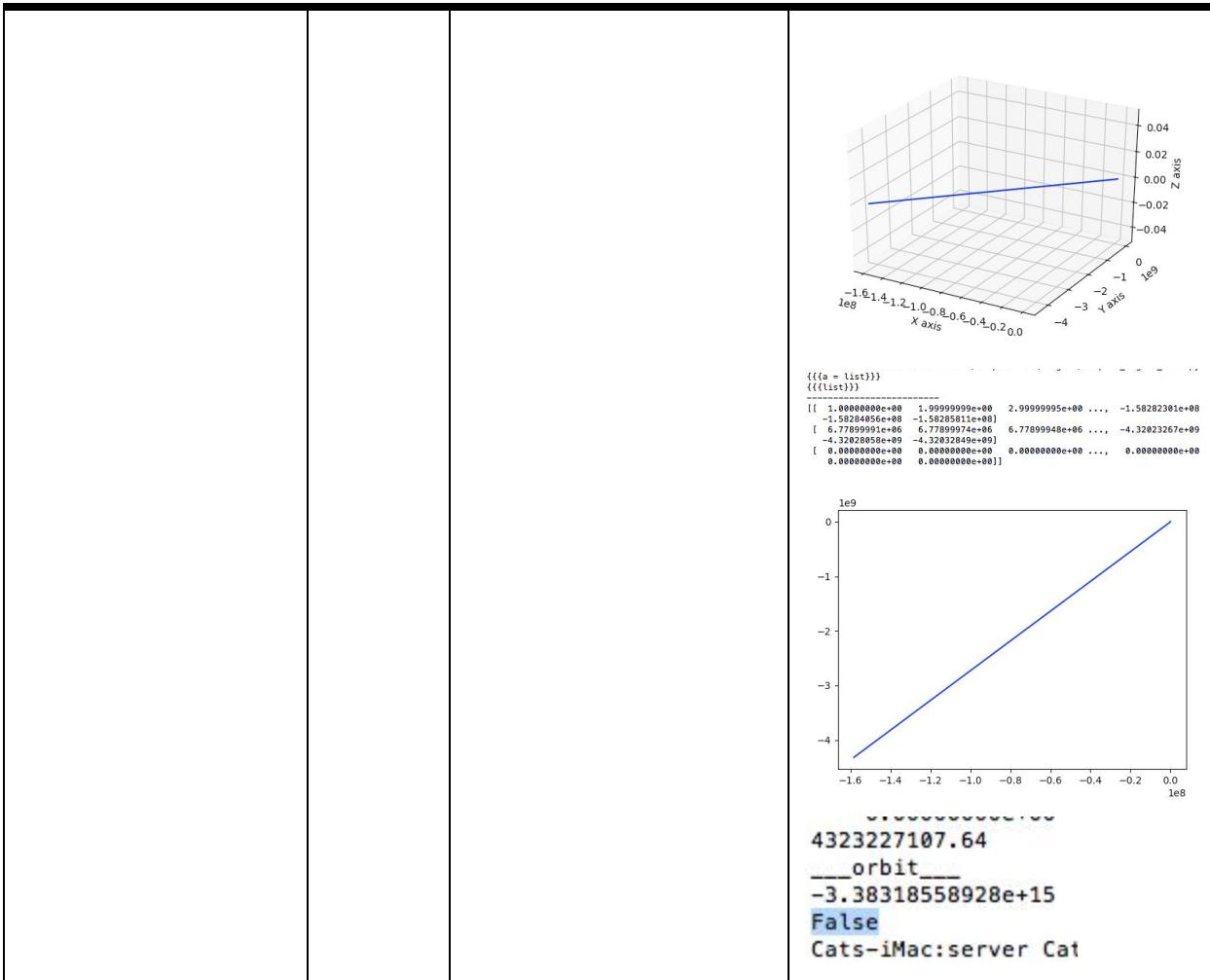
```
.....  
7096266.69238  
__orbit__  
1884185.89591  
True  
Cats-iMac:server
```



```
.....  
4323227107.64  
__orbit__  
-3.38318558928e+15  
False  
Cats-iMac:server Cat
```

150318E1

Test	Pass/Fail	Description	Evidence
Check that orbit checking function works	PASS	<p>Variables for newtonian orbit should make the orbit check function return true. The XYZ graph should confirm this.</p> <p>Variables for non-orbit should make the orbit check function return false. The XYZ should confirm this. This check is important to make sure that the only current game objective works as needed.</p>	<pre># In[]: # ---- SIM > Orbit (hopefully) saturnv.mass = 500 saturnv.a = [0, 0, 0] saturnv.F = 0 saturnv.pos = [0, 6779e3, 0] # 408km above Earth's surface saturnv.V = [7800, 0, 0]</pre>  <pre>Cats-iMac:server Cats:~ python3 /Users/Ben/Desktop/CS/Testing/150318E1/mission_cont_03e2727a0de646cc1baa436dc38f2/Computation_Engine/Compute_engine_test.py [[{"a": 100000000}], [{"list"}] [[{"list"}]] [[{"7.0000000e+02 1.5590000e+03 2.3399999e+03 ..., -6.70074255e+06 -6.70096446e+06 -6.70118629e+06 6.7789991e+06 6.77899974e+06 6.77899948e+06 ..., -2.33618329e+06 -2.33547155e+06 -2.33475978e+06 0.8000000e+00 0.8000000e+00 0.0000000e+00 ..., 0.0000000e+00 0.0000000e+00}]]</pre>  <pre># In[]: # ---- SIM > Orbit (hopefully) saturnv.mass = 500 saturnv.a = [0, 0, 0] saturnv.F = 0 saturnv.pos = [0, 6779e3, 0] # 408km above Earth's surface saturnv.V = [10, 0, 0]</pre> <pre>7096266.69238 __orbit__ 1884185.89591 True Cats-iMac:server</pre>



Working Computational Engine

Browse files

I now have a minimum viable solution for my computational engine, where it correctly calculates the correct equations needed for the program to run. I've made sure it has been made in a way that enables me to be able to add more functionality to the code at a later date, with the use of functions and an object oriented design.

Currently the code is independent and runs itself however from line 157 onward will be removed as the run function will be called iteratively from the server.

↳ experiments

 meowterspace committed 23 days ago Verified

1 parent 2073659 commit d56c577dfba4c2411f405b5b950602b1e68125ea

Showing 1 changed file with 156 additions and 52 deletions.

Unified Split

208 3 0 0 Computation Engine/resources.py

```

...
...
@@ -1,77 +1,181 @@
1   1   import numpy as np
2   2   +import matplotlib.pyplot as plt
3   3   +from mpl_toolkits.mplot3d import Axes3D
4
5   -objects = [] # a list of all the objects instantiated into the simulation
6   +OBJECTS = [] # a list of all the objects instantiated into the simulation
7   +
8   +def resolve(vector):
9   +    r = np.sqrt((vector[0]**2)+(vector[1]**2)+(vector[2]**2))
10
11  10  class GameObject():
12  -    def __init__(self, mass, radius, pos, V, a):
13  -        self.mass, self.radius = mass, radius
14  -        self.pos, self.V, self.a = pos, V, a
15
16  -    def distance(self, obj):
17  -        d = [None, None, None]
18  -        d[0] = self.pos[0]-obj.pos[0]
19  -        d[1] = self.pos[1]-obj.pos[1]
20  -        d[2] = self.pos[2]-obj.pos[2]
21  -        r = np.sqrt((d[0]**2)+(d[1]**2)+(d[2]**2))
22  -        return r
23  +    def __init__(self, mass, radius, pos, V, a):
24  +        self.mass, self.radius = mass, radius
25  +        self.pos, self.V, self.a = pos, V, a
26
27  +    def distance(self, obj):
28  +        d = [None, None, None]
29  +        d[0] = self.pos[0]-obj.pos[0]
30  +        d[1] = self.pos[1]-obj.pos[1]
31  +        d[2] = self.pos[2]-obj.pos[2]
32  +        r = np.sqrt((d[0]**2)+(d[1]**2)+(d[2]**2))
33  +        return r
34
35  23  def gravity(self, obj):
36  -    F = (6.67408e-11 * obj.mass * self.mass) / (self.distance(obj)**2)
37  -    return F
38  +    F = (6.67408e-11 * obj.mass * self.mass) / (self.distance(obj)**2)
39  +    return F
40
41  27  def get_grav_vector(self, obj):
42  -    Fgrav = gravity(obj.mass, self.mass, self.get_home_radius())
43  -    Fgrav = self.gravity(obj)
44  -    Fgravx = self.pos[0]/self.distance(obj)*Fgrav
45  -    Fgravy = self.pos[1]/self.distance(obj)*Fgrav
46  -    Fgravz = self.pos[2]/self.distance(obj)*Fgrav
47  -    return [Fgravx, Fgravy, Fgravz]
48
49  -
```



```

56      86         return [x, y, z]
57
58      - def thrust(self): # q = rate of ejected mass flow, Ve = exhaust gas ejection speed
59      -     F = q * Ve + (Pe - Pa) * Ae # Pe = pressure of exhaust gasses, Pa = pressure of ambient atmosphere
60      -     return F # Ae = area of exit
61
62      - def drag_area():
63      -     pass
64      + def thrust(self, planet): # q = rate of ejected mass flow, Ve = exhaust gas ejection speed
65
66      - def drag(planet, T, h): # P = Pressure /PA, T = temp(k), Vg = flow velocity of gas
67      -     F = 0.5 * ((planet.pressure(T, h) / (286 * T)) * (self.Vg ** 2) * self.Cd * drag_area()) # Cd = Coefficient of
68      -     return F
69
70      -
71
72      -
73
74      -
75      -Earth = Planet(5.972e24, 6371e3, 101325, 0.02896, [0, 0, 0])
76      -zero = [0, 0, 0]
77      +     F = self.q * self.Ve + (self.Pe - planet.atmosphere.pressure(285, self.distance(planet)-planet.radius)) * self.#
78      +
79      +     return F # Ae = area of exit
80
81      + def drag_area():
82      +     return 1
83      +
84      + def drag(planet, T, h): # P = Pressure /PA, T = temp(k), Vg = velocity
85      +     F = 0.5 * ((planet.pressure(T, h) / (286 * T)) * (self.V ** 2) * self.Cd * self.drag_area()) # Cd = Coefficien
86      +     return F
87
88      +
89      + def die():
90      +     return True
91
92      +
93
94      +
95
96      +
97      +
98      +
99      +
100
101
102
103
104
105
106
107
108
109
110
111
112 +def make_planet(Planet, name, mass, radius, pos, V, a, p0, molMass):
113 +    exec(str(name)+ = Planet('+str(mass)+','+str(radius)+','+str(pos)+','+str(V)+','+str(a)+','+p0+','+str(molMass)+'
114 +    exec('OBJECTS.append( ['+str(name)+', "planet"] )')
115
116
117
118 +# funct:
119 +def setup():
120 +    global OBJECTS
121 +    #make_planet(Planet, 'Earth', 5.972e24, 6371e3, [0, 0, 0], [0, 0, 0], [0, 0, 0], 101325, 0.02896)
122 +    Earth = Planet(5.972e24, 6371e3, [0, 0, 0], [0, 0, 0], [0, 0, 0], 101325, 0.02896)
123 +    OBJECTS.append([Earth, 'planet'])
124 +    player = Rocket(100, 3, 0.5, 0, [6e24, 0, 0], [90, 0, 0], [0, 0, 0], [0, 0, 0], 30, 3100, 5000, 0.7, 50, 2, 10000) ;
125 +    OBJECTS.append([player, 'player'])
126 +    return player
127
128
129 +data = []
130
131 +def run(planet, player, response_t):
132 +    global OBJECTS
133 +    print(player.pos)
134 +    F = [0, 0, 0]
135 +    player.F = [0, 0, 0]
136 +    player.F = player.thrust(planet)
137 +    F = player.resolve_thrust()
138 +    for i in OBJECTS:
139 +        if i[1] == 'planet':

```

```

138     #     F = F - player.drag(i[0], 285, player.distance(i[0] - i[0].radius)) # TEMPERATURE SET TO CONST 285K
139     F = [F[0]+player.get_grav_vector(i[0])[0], F[1]+player.get_grav_vector(i[0])[1],F[2]+player.get_grav_vector(i[0])[2]]
140
141     if (i != 'player'):
142         player.collision_test(i[0])
143     player.F = F
144     player.a = [player.F[0]/player.mass, player.F[1]/player.mass, player.F[2]/player.mass] # a = f/m
145     u = player.V
146     player.V = [player.V[0]+player.a[0]*response_t, player.V[1]+player.a[1]*response_t, player.V[2]+player.a[2]*response_t]
147     s = [(u[0]*response_t) + (0.5*player.a[0]*(response_t**2)), (u[1]*response_t) + (0.5*player.a[1]*(response_t**2)), (u[2]*response_t) + (0.5*player.a[2]*(response_t**2))]
148     player.pos = [player.pos[0]+s[0], player.pos[1]+s[1], player.pos[2]+s[2]]
149
150     print('a = '+str(player.a))
151     print('v = '+str(player.V))
152     print('x = '+str(player.pos[0]))
153     print('y = '+str(player.pos[1]))
154     print('z = '+str(player.pos[2]))
155     data.append(player.pos)
156
157     +fig = plt.figure()
158     +ax = plt.axes(projection='3d')
159
160     ## Create All objects E.g. Planets \/
161     ## Create rocket \/
162     ## Set up rocket \/
163     ## make game.time variable (probably in main run script)
164     ## funct:
165     ## run calculations:
166     ## STA Drag 3 - LAST
167     ## STA Thrust 1
168     ## STA Grav 2
169     ## STA
170
171     +Player = setup()
172     +print('ok')
173     +for object in OBJECTS:
174         if object[1] == 'planet':
175             for i in range(100):
176                 print(i)
177                 run(object[0], Player, 1)
178
179     +data = np.array(data)
180     +data = np.transpose(data)
181     +ax.plot(data[0], data[1], data[2])

```

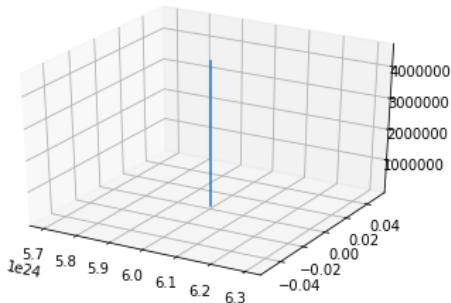
1 comment on commit [d56c577](#)



meowterspace commented on d56c577 23 days ago

Owner

Matplotlib output:

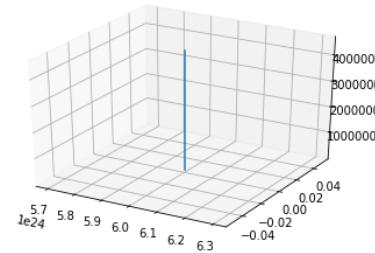


a graph showing the positions of the rocket

Evidently my rocket flies in a straight line which from the runcode should be correct

I've outputted this graph to get a visual representation of the numbers above so i can check that it's working correctly

230318E1

Test	Pass/Fail	Description	Evidence
Check resources.py and all its functions work together as expected	PASS	Resources.py should run with no errors and the produced key variables should match with the XYZ position graph	<pre>[5.99999999999999e+24, 0.0, 1841.6331270171493] a = [-461.51582461304525, 0.0, 920.81656350857463] v = [-1384.5474738391358, 0.0, 2762.4496905257238] x = 6e+24 y = 0.0 z = 4143.67453579 3 [5.99999999999999e+24, 0.0, 4143.6745357885857] a = [-461.51582461304525, 0.0, 920.81656350857463] v = [-2307.063298452181, 0.0, 3683.2662548342985] x = 6e+24 y = 0.0 z = 7366.53250807 4 [5.99999999999999e+24, 0.0, 7366.53250808685961] a = [-461.51582461304525, 0.0, 920.81656350857463] v = [-2307.5791230652262, 0.0, 4604.0828175428733] x = 6e+24 y = 0.0 z = 11510.2070439 5 [5.99999999999999e+24, 0.0, 11510.207043857183] a = [-461.51582461304525, 0.0, 920.81656350857463] v = [-2769.094947682716, 0.0, 5524.8993810514476] x = 6e+24 y = 0.0 z = 16574.6981432 6 [5.99999999999999e+24, 0.0, 16574.698143154343] a = [-461.51582461304525, 0.0, 920.81656350857463] v = [-3230.6107722913171, 0.0, 6445.7159445600219] x = 6e+24 y = 0.0 z = 22560.005806 7 [5.99999999999999e+24, 0.0, 22560.005805960078] a = [-461.51582461304525, 0.0, 920.81656350857463] v = [-3692.1265969043625, 0.0, 7366.53250808685961] x = 6e+24 y = 0.0 z = 29466.1300323 8 [5.99999999999999e+24, 0.0, 29466.130032274388] a = [-461.51582461304525, 0.0, 920.81656350857463] v = [-4153.6424215174075, 0.0, 8287.3490715771713] x = 6e+24 y = 0.0 z = 37293.0708221 9 [5.99999999999999e+24, 0.0, 37293.070822097274] a = [-461.51582461304525, 0.0, 920.81656350857463] v = [-4615.1582461304524, 0.0, 9208.1656350857465] x = 6e+24</pre> 

Linked files

Browse files

resources.py is now called from app.py (run function). This will enable the calculations to be run every time a message is received to keep the data flow streaming to the client.

The computational(game) data is packaged as an array along with meta data, which is used to identify the data package uniquely. This is important for tracking packet loss at a later date, checking if a client is still connected and timing the interval between data being sent to the client and data being sent back from the client (ping response time).

` master

meowterspace committed 15 days ago

1 parent c0a71ee commit 4308f0eb2e3b41dec4ab720e6cf3e655aa1cde72

Showing 2 changed files with 41 additions and 16 deletions.

Unified Split

36 client/ui.html

```

20 20
21 21
22 22
23 23
24 24
25 25
26 26
27 27
28 28
29 29
30 30
31 31
32 32
33 33
34 34
35 35
36 36
37 37
38 38
39 39
40 40
41 41
42 42
43 43
44 44
45 45
46 46
47 47
48 48
49 49
50 50
51 51
52 52
53 53
54 54
55 55
56 56
57 57
58 58
59 59
60 60
61 61
62 62
63 63
64 64
65 65
66 66
67 67
68 68
69 69
70 70
71 71
72 72
73 73
74 74
75 75
76 76
77 77
78 78
79 79
80 80
81 81
82 82
83 83
84 84
85 85
86 86
87 87
88 88
89 89
90 90
91 91
92 92
93 93
94 94
95 95
96 96
97 97
98 98
99 99
100 100
101 101
102 102
103 103
104 104
105 105
106 106
107 107
108 108
109 109
110 110
111 111
112 112
113 113
114 114
115 115
116 116
117 117
118 118
119 119
120 120
121 121
122 122
123 123
124 124
125 125
126 126
127 127
128 128
129 129
130 130
131 131

```

height: 100%;

}

#win1 {

width: 200px;

height: 400px;

width: 400px;

height: 600px;

}

#win-FDAI {

width: 400px;

var win = document.getElementById(id);

\$(win).css("z-index", ++zIndex);

\$(win).show();

if (id=='win-fdai' && FDAI_rendered == 0) {

if (id=='win-fdai' && FDAI_rendered >= 0) {

render_navball();

FDAI_rendered = 1;

}

// UPDATE PAGE DATA HERE

//document.getElementById('time').innerHTML = 'Time ('+msg.zone+'): '+msg.time;

var readout;

document.getElementById('serv').innerHTML = msg.serv;

cube.rotation.x = msg.angle[0];

readout = readout + '<p>p_acc: '+String(msg.p_acc)+'</p>';

readout = readout + '<p>p_ang: '+String(msg.p_ang)+'</p>';

readout = readout + '<p>p_fue: '+String(msg.p_fue)+'</p>';

readout = readout + '<p>p_orb: '+String(msg.p_orb)+'</p>';

readout = readout + '<p>p_pos: '+String(msg.p_pos)+'</p>';

readout = readout + '<p>p_sta: '+String(msg.p_sta)+'</p>';

readout = readout + '<p>p_thm: '+String(msg.p_thm)+'</p>';

readout = readout + '<p>p_vel: '+String(msg.p_vel)+'</p>';

readout = readout + '<p>serv: '+String(msg.serv)+'</p>';

readout = readout + '<p>time: '+String(msg.time)+'</p>';

readout = readout + '<p>uuid: '+String(msg.uuid)+'</p>';

readout = readout + '<p>zone: '+String(msg.zone)+'</p>';

document.getElementById('readout').innerHTML = readout;

+

document.getElementById('serv').innerHTML = msg.serv;

if (FDIAI_rendered == 1) {

FDIAI_rendered = FDIAI_rendered + 1;

} else if (FDIAI_rendered ==2) {

cube.rotation.x = msg.angle[0];

};

socket.send(msg.uuid);

- console.log(msg.uuid);

+ console.log(msg);

```

112 132    });
113 133
114 134  });
132 152      </div>
133 153  </nav>
134 154  <div id="win1" class="col-3 panel panel-default">
135  -      <div class="handle panel-heading">Window 1
136  +      <div class="handle panel-heading">readout
137 156          <button type="button" class="close" aria-label="Close" onclick="hideModal('win1')">
138 157              <span aria-hidden="true">&times;</span>
139 159          </button>
140 160      </div>
141 163  </div>
142 164
143 165  <div id="win-fdai" class="col-2 panel panel-danger">

```

21 server/app.py

```

7  7  from flask import Flask, render_template, send_file, request, session, redirect, render_template_string
8  8  from flask_socketio import SocketIO, emit, send
9  9  from threading import Thread
10 +import resources
11
12 12  header = ''
13 13  <head>
119 120  socketio = SocketIO(app)
120
121 122
122 -data = {
123 +meta = {
124     'time' : '',      #current Time
125     'zone' : 'GMT',   #timezone
126     'serv' : '',      #time the server's been active
127 -     'uuid' : '',    #unique message id
127 +     'uuid' : ''     #unique message id
128 128 }
129
130 130  start = datetime.datetime.now()
180 180  @socketio.on('message')
181 181  def handle_message(message):
182
183 -     if message == str(data['uuid']): print(message)
183 +     if message == str(meta['uuid']): print(message)
184     else: print('Packet Loss!')
185
186 -     data['time'] = str(datetime.datetime.now())
187 -     data['serv'] = str(datetime.datetime.now()-start)
188 -     data['uuid'] = str(uuid.uuid4())
189 -     data['angle'][0] = data['angle'][0]+0.01
190 -     send(data)
186 +     meta['time'] = str(datetime.datetime.now())
187 +     meta['serv'] = str(datetime.datetime.now()-start)
188 +     meta['uuid'] = str(uuid.uuid4())
189 +
190 +     to_send = {}
191 +     to_send.update(meta)
192 +     to_send.update(resources.data)
193 +     send(to_send)
191
192
193 196  @socketio.on('message', namespace='/lobbu')

```

0 comments on commit 4308f0e

Linked Files Properly

[Browse files](#)

In my last commit, I wasn't actually running the calculations. In this commit the code is actually being run, as well as the computation being set up.

by master

 meowterspace committed 15 days ago

1 parent 4308f0e commit 08e38983783c1163a2c4280156faeb611a4de527

Showing 2 changed files with 215 additions and 1 deletion.

[Unified](#) [Split](#)

12 server/app.py

```
90  90  }
91  91  }
92  92  users = []
93  93  -
94  93  class User:
95  94      def __init__(self, name):
96  95          self.name = name
129 128
130 129  start = datetime.datetime.now()
131 130
131 131  +===== RUN =====
132 132  +player = resources.setup()
133 133  +
132 134  #===== APP ROUTES - FLASK =====
133 135
134 136  @app.route('/')
177 179
178 180  #===== APP ROUTES - SOCKETIO =====
179 181
182 182  +
180 183  @socketio.on('message')
181 184  def handle_message(message):
185 185      print('setup')
186 186      for i in resources.OBJECTS:
187 187          if (i[1] == 'planet'):
188 188              resources.run(i[0], player, 0.1)
189 189
190 190  if message == str(meta['uuid']): print(message)
191 191  else: print('Packet Loss!')
192 199  to_send.update(resources.data)
193 200  send(to_send)
194 201
202 202  +@socketio.on('message', namespace='/edit_data')
203 203  +def handle_incoming_data(message):
204 204      print('Incoming Data: '+message)
195 205
196 206  +@socketio.on('message', namespace='/lobbu')
197 207  def handle_lobby_message(message):
```

204 server/resources.py

```
...
...
@@ -0,0 +1,204 @@
+import numpy as np
+
+OBJECTS = [] # a list of all the objects instantiated into the simulation
+
+data = {
+    'p_acc' : [None, None, None],    #player acceleration
+    'p_vel' : [None, None, None],    #player velocity
+    'p_pos' : [None, None, None],    #player position
+    'p_ang' : [None, None, None],    #player angle
+    'p_fue' : '',                  #player fuel           <<<< ADD IN TO GAME
+    'p_thm' : 0,                   #player thrust multiplier <<<< ADD IN TO GAME
```

```

12  + 'p_sta' : True,      #player status           <<<< ^
13  + 'p_orb' : False,     #player orbit
14  +
15  +
16  +def resolve(vector):
17  +    r = np.sqrt((vector[0]**2)+(vector[1]**2)+(vector[2]**2))
18  +
19  +class GameObject():
20  +    def __init__(self, mass, radius, pos, V, a):
21  +        self.mass, self.radius = mass, radius
22  +        self.pos, self.V, self.a = pos, V, a
23  +
24  +    def distance(self, obj):
25  +        d = [None, None, None]
26  +        d[0] = self.pos[0]-obj.pos[0]
27  +        d[1] = self.pos[1]-obj.pos[1]
28  +        d[2] = self.pos[2]-obj.pos[2]
29  +        r = np.sqrt((d[0]**2)+(d[1]**2)+(d[2]**2))
30  +        return r
31  +
32  +    def gravity(self, obj):
33  +        F = (6.67408e-11 * obj.mass * self.mass) / (self.distance(obj)**2)
34  +        return F
35  +
36  +    def get_grav_vector(self, obj):
37  +        Fgrav = self.gravity(obj)
38  +        Fgravx = self.pos[0]/self.distance(obj)*Fgrav
39  +        Fgravy = self.pos[1]/self.distance(obj)*Fgrav
40  +        Fgravz = self.pos[2]/self.distance(obj)*Fgrav
41  +        return [Fgravx, Fgravy, Fgravz]
42  +
43  +    def collision_test(self, object): #rocket, planet
44  +        if (self.length > self.radius):
45  +            brad = self.length
46  +        else:
47  +            brad = self.radius
48  +        d = self.distance(object) - brad - object.radius
49  +        if (d == 0):
50  +            if (np.mod(resolve(self.V)-resolve(object.V)) < 10):
51  +                velocity_ratio = [None, None, None]
52  +                velocity_ratio[0] = self.v[0] / resolve(self.V)
53  +                velocity_ratio[1] = body.v[1] / resolve(self.V)
54  +                velocity_ratio[2] = body.v[2] / resolve(self.V)
55  +                pb = self.mass * resolve(self.V)
56  +                po = object.mass * resolve(object.V)
57  +                vr = (pb + po) / (object.mass + self.mass)
58  +                self.V = [vr*velocity_ratio[0], vr*velocity_ratio[1], vr*velocity_ratio[2]]
59  +                object.V = [vr*velocity_ratio[0], vr*velocity_ratio[1], vr*velocity_ratio[2]]
60  +            elif (d < 0):
61  +                #die
62  +                pass
63  +            else:
64  +                # die
65  +                pass
66  +
67  +
68  +class Planet(GameObject):
69  +    class Atmosphere(GameObject):
70  +        def __init__(self, mass, radius, pos, V, a, p0, molMass):
71  +            super().__init__(mass, radius, pos, V, a)
72  +            self.mass, self.p0, self.molMass = mass, p0, molMass
73  +
74  +        def pressure(self, T, h):
75  +            g = (self.mass*6.67408e-11)/(self.radius+h) # check this works, i'm a bit dubious
76  +            P = self.p0 * np.exp((-1 * (self.molMass * g) / (8.3145 * T)) * h)
77  +            return P
78  +
79  +        def __init__(self, mass, radius, pos, V, a, p0, molMass):
80  +            super().__init__(mass, radius, pos, V, a)
81  +            atmosphere = self.Atmosphere(mass, radius, pos, V, a, p0, molMass)

```

```

82     self.atmosphere = atmosphere
83
84 +
85 +class Rocket(GameObject):
86 +    def __init__(self, mass, length, radius, F, pos, angle, V, a, q, Ve, Pe, Cd, Vg, Ae, Fuel):
87 +        super().__init__(mass, radius, pos, V, a)
88 +        self.length, self.F, self.angle = length, F, angle
89 +        self.q, self.Ve, self.Pe, self.Ae = q, Ve, Pe, Ae
90 +        self.Cd, self.Vg, self.fuel = Cd, Vg, Fuel
91 +    def resolve_thrust(self): # yaw, pitch, roll -> x, y, z
92 +        x = self.F*(np.cos(self.angle[0])*np.cos(self.angle[1]))
93 +        y = self.F*(np.sin(self.angle[1]))
94 +        z = self.F*(np.sin(self.angle[0])*np.cos(self.angle[1]))
95 +        return [x, y, z]
96 +
97 +    def thrust(self, planet): # q = rate of ejected mass flow, Ve = exhaust gas ejection speed
98 +
99 +        F = self.q * self.Ve + (self.Pe - planet.atmosphere.pressure(285, self.distance(planet)-planet.radius)) * self.fuel
100 +
101 +        return F # Ae = area of exit
102 +
103 +    def drag_area():
104 +        return 1
105 +
106 +    def drag(planet, T, h): # P = Pressure /PA, T = temp(k), Vg = velocity
107 +        F = 0.5 * ((planet.pressure(T, h) / (286 * T)) * (self.V ** 2) * self.Cd * self.drag_area()) # Cd = Coefficient of drag
108 +        return F
109 +
110 +    def die():
111 +        return True
112 +
113 +
114 +
115 +
116 +
117 +
118 +
119 +
120 +
121 +def make_planet(Planet, name, mass, radius, pos, V, a, p0, molMass):
122 +    exec(str(name)+' = Planet('+str(mass)+','+str(radius)+','+str(pos)+','+str(V)+','+str(a)+','+str(p0)+','+str(molMass)+')
123 +    exec('OBJECTS.append( ['+str(name)+', "planet"] )')
124 +
125 +
126 +## funct:
127 +def setup():
128 +    global OBJECTS
129 +    #make_planet(Planet, 'Earth', 5.972e24, 6371e3, [0, 0, 0], [0, 0, 0], [0, 0, 0], 101325, 0.02896)
130 +    Earth = Planet(5.972e24, 6371e3, [0, 0, 0], [0, 0, 0], [0, 0, 0], 101325, 0.02896)
131 +    OBJECTS.append([Earth, 'planet'])
132 +    player = Rocket(100, 3, 0.5, 0, [6e24, 0, 0], [90, 0, 0], [0, 0, 0], [0, 0, 0], 30, 3100, 5000, 0.7, 50, 2, 10000)
133 +    OBJECTS.append([player, 'player'])
134 +    return player
135 +
136 +
137 +
138 +def run(planet, player, response_t):
139 +    global OBJECTS
140 +    global data
141 +    print(player.pos)
142 +    F = [0, 0, 0]
143 +    player.F = [0, 0, 0]
144 +    player.F = player.thrust(planet)
145 +    F = player.resolve_thrust()
146 +    for i in OBJECTS:
147 +        if i[1] == 'planet':
148 +            # F = F - player.drag(i[0], 285, player.distance(i[0] - i[0].radius)) # TEMPERATURE SET TO CONST 285K
149 +            F = [F[0]+player.get_grav_vector(i[0])[0], F[1]+player.get_grav_vector(i[0])[1], F[2]+player.get_grav_vector(i[0])[2]]
150 +
151 +        if (i != 'player'):
```

```

152     +         player.collision_test(i[0])
153     +     player.F = F
154     +     player.a = [player.F[0]/player.mass, player.F[1]/player.mass, player.F[2]/player.mass] # a = f/m
155     +     u = player.V
156     +     player.V = [player.V[0]+player.a[0]*response_t, player.V[1]+player.a[1]*response_t, player.V[2]+player.a[2]*response_t]
157     +     s = [(u[0]*response_t) + (0.5*player.a[0]*(response_t**2)), (u[1]*response_t) + (0.5*player.a[1]*(response_t**2)), (u[2]*response_t) + (0.5*player.a[2]*(response_t**2))]
158     +     player.pos = [player.pos[0]+s[0], player.pos[1]+s[1], player.pos[2]+s[2]]
159
160     +     print('a = '+str(player.a))
161     +     print('v = '+str(player.V))
162     +     print('x = '+str(player.pos[0]))
163     +     print('y = '+str(player.pos[1]))
164     +     print('z = '+str(player.pos[2]))
165
166     +     data['p_acc'] = player.a
167     +     data['p_vel'] = player.V
168     +     data['p_pos'] = player.pos
169     +     data['p_ang'] = player.angle
170     +     data['p_fue'] = 100 # ADD THIS
171     +     data['p_thm'] = 100 # ADD THIS
172     +     data['p_sta'] = True # ADD THIS
173     +     data['p_orb'] = False # ADD THIS
174
175
176     +def update_data(data):
177     +     global OBJECTS
178     +     player.a = data['p_acc']
179     +     player.V = data['p_vel']
180     +     player.pos = data['p_pos']
181     +     player.angle = data['p_ang']
182     +     #FUE, THM, STA, ORB
183
184
185
186     +# Create All objects E.g. Planets \
187     +# Create rocket \
188     +# Set up rocket \
189     +# make game.time variable (probably in main run script)
190     +# funct:
191     +# run calculations:
192     +# STA Drag 3 - LAST
193     +# STA Thrust 1
194     +# STA Grav 2
195     +# STA
196
197     +Player = setup()
198     +print('ok')
199     +for object in OBJECTS:
200     +     if object[1] == 'planet':
201     +         for i in range(100):
202     +             print(i)
203     +             run(object[0], Player, 1)
204     +             0#

```

0 comments on commit 08e3898

Numerical Figure Selector

[Browse files](#)

I've built a small Javascript UI item that allows you to change the value of a variable by increments of a power of 10. E.g. 10, 1, 0.1 using up and down arrows.

I have done this because it allows more accurate control of variables such as thrust and is very easy to understand for first time users. It follows my design exactly and should make the game more interactive and easy.

↗ experiments

 meowterspace committed 14 days ago

1 parent d56c577 commit d3c56541bb5770df3ca6d67e9c6d5e561fbf48f2

Showing 3 changed files with 87 additions and 0 deletions.

[Unified](#) [Split](#)

19 UI/numerical figure selector/css/style.css

```
... | ... @@ -0,0 +1,19 @@
1 | +td{
2 | +  text-align: center;
3 | +}
4 | +
5 | +input{
6 | +  text-align: left;
7 | +}
8 | +
9 | +#box{
10 | +  width: 200px;
11 | +  height: 200px;
12 | +  background-color: #f1f1f1;
13 | +}
14 | +
15 | +#up{
16 | +  width: 20px;
17 | +  height: 20px;
18 | +  background-color: #ffffdd0;
19 | +}
```

51 UI/numerical figure selector/index.html

```
... | ... @@ -0,0 +1,51 @@
1 | +<!DOCTYPE html>
2 | +<html lang="en" >
3 | +<script src="js/index.js"></script>
4 | +<head>
5 | +  <meta charset="UTF-8">
6 | +  <title>An Anonymous Pen on CodePen</title>
7 | +
8 | +
9 | +
10 | +  <link rel="stylesheet" href="css/style.css">
11 | +
12 | +
13 | +</head>
14 | +
15 | +<body>
16 | +
17 | +  <div id='box'>
18 | +    <table>
19 | +      <tr>
20 | +        <td><input type='button' onclick="change(10, 1)" id='up' value='▲'></td>
21 | +        <td><input type='button' onclick="change(1, 1)" id='up' value='▲'></td>
22 | +        <td><input type='button' id='up'></td>
23 | +        <td><input type='button' onclick="change(0.1, 1)" id='up' value='▲'></td>
24 | +      </tr>
25 | +      <tr id='text'>
```

```

26  +      <td id='thm1'>2</td>
27  +      <td id='thm2'>5</td>
28  +      <td id='thm3'>.</td>
29  +      <td id='thm4'>3</td>
30  +    </tr>
31  +  <tr>
32  +    <td><input type='button' onclick="change(10, -1)" id='up' value='▼'></td>
33  +    <td><input type='button' onclick="change(1, -1)" id='up' value='▼'></td>
34  +    <td><input type='button' id='up'></td>
35  +    <td><input type='button' onclick="change(0.1, -1)" id='up' value='▼'></td>
36  +  </tr>
37  + </table>
38  +</div>
39  +
40  +
41  +
42  +
43  +
44  +
45  +
46  +
47  +
48  +
49  +</body>
50  +
51  +</html>

```

17 UI/numerical figure selector/js/index.js

```

...
...
@@ -0,0 +1,17 @@
+var value = 25.3;
+
+
+function make(value){
+  value = value.toString();
+  value = value.split('');
+  document.getElementById('thm1').innerHTML = value[0];
+  document.getElementById('thm2').innerHTML = value[1];
+  document.getElementById('thm3').innerHTML = value[2];
+  document.getElementById('thm4').innerHTML = value[3];
+};
+
+function change(power, direction){
+  value = value + (1*power*direction);
+  console.log(value);
+  make(value);
+}; o*

```

1 comment on commit d3c5654



meowterspace commented on d3c5654 3 minutes ago

Owner

▲	▲	▼	▲
1	0	6	0
▼	▼	▼	▼

▲	▲	▼	▲
2	6	.	0
▼	▼	▼	▼

▲	▲	▼	▲
-	3	.	9
▼	▼	▼	▼

▲	▲	▼	▲
6	.	0	9
▼	▼	▼	▼

▲	▲	▼	▲
1	0	6	.
▼	▼	▼	▼

▲	▲	▼	▲
2	6	undefined	undefined
▼	▼	▼	▼

35.30000000000004
 35.40000000000006
 35.50000000000001
 35.40000000000006
 35.30000000000004
 35.2
 35.30000000000004
 36.30000000000004
 26.30000000000004

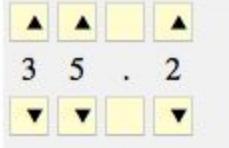
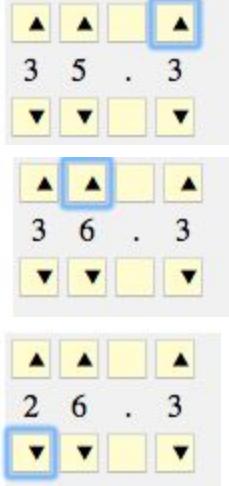
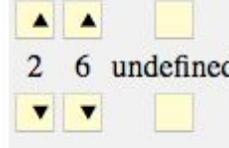
▲	▲	▼	▲
2	6	.	3
▼	▼	▼	▼

▲	▲	▼	▲
3	6	.	3
▼	▼	▼	▼

▲	▲	▼	▲
3	5	.	3
▼	▼	▼	▼

▲	▲	▼	▲
3	5	.	2
▼	▼	▼	▼

010418E1

Test	Pass/Fail	Description	Evidence
Check numerical variable editors display correctly	PASS	The NVE should display correctly with arrow buttons aligned correctly over each digit. It is important to check this looks right before continuing so it won't be confusing for users.	
Check numerical variable editors can change a specific digit by a power of 10	PASS	The corresponding digit should change up or down by 1 to make sure that the NVE works as it should.	
Check JS records correct variable to match numerical editor	PASS	The number printed in the JS console should match the number on screen precisely to make sure it can be sent to the server correctly	<p>35.3000000000000004 35.4000000000000006 35.5000000000000001 35.4000000000000006 35.3000000000000004 35.2 35.3000000000000004 36.3000000000000004 26.3000000000000004</p>
Check numerical variable editor can handle 0 values	FAIL	The digit that is 0 should show the 0 character and not Null, Non, NaN, or Undefined to make sure it displays correctly.	

Check numerical variable editor can't go outside it's range	FAIL	The NVE shouldn't overflow outside its boundaries. The decimal point shouldn't move, instead the value shouldn't be allowed to be increased over 100 to make sure it displays correctly.	
Check numerical variable editor can handle change in decimal point position	FAIL	The NVE should not move the decimal point to compensate but instead a 0 should be added to the start of the value to make it easier to read and avoid user confusion.	
Check numeric variable editor can handle negative values	FAIL	The NVE should add a negative sign at the beginning instead of changing one of the digits to a negative sign to avoid confusion for the end user.	

Update numerical figure selector

Browse files

I've updated the NFS UI so that the javascript function is generic and applies to any function. This is important because I don't have to re-write the function many times over and is reusable. This can be seen in the `make()` function.

It can be altered by passing in a list of the html IDs that it should be altering, the value to change it to and where the decimal point should lie.

💡 experiments

 meowterspace committed 14 days ago

1 parent d3c5654 commit 22694e8f53f31aa51160058a717a8a05e79b50f6

 Showing 1 changed file with 9 additions and 6 deletions.

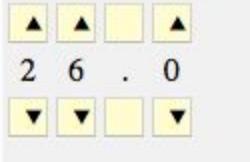
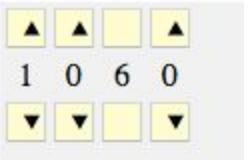
Unified Split

15 UI/numerical_figure_selector/js/index.js

```
... @@ -1,17 +1,20 @@
1 1 var value = 25.3;
2
3
4 -function make(value){
5 +function make(id, value, dp){ // List of IDs for values, value, decimal point position in array
6 +    value = Math.round(value*100)/100;
7 +    console.log(value);
8     value = value.toString();
9     value = value.split('');
10 -    document.getElementById('thm1').innerHTML = value[0];
11 -    document.getElementById('thm2').innerHTML = value[1];
12 -    document.getElementById('thm3').innerHTML = value[2];
13 -    document.getElementById('thm4').innerHTML = value[3];
14 +    for (var i=0; i<id.length; i++){
15 +        if (value[i] == null){ value[i] = '0';};
16 +        if (value[dp] == null){ value[dp] = '.';};
17 +        document.getElementById(id[i]).innerHTML = value[i];
18 +    };
19 +    make(['thm1', 'thm2', 'thm3', 'thm4'], value, 2);
20 }; @@
```

0 comments on commit 22694e8

010418E2

Test	Pass/Fail	Description	Evidence
Check numerical variable editor can handle 0 values	PASS	The digit that is 0 should show the 0 character and not Null, Non, NaN, or Undefined to make sure it displays correctly.	
Check numerical variable editor can't go outside it's range	FAIL	The NVE shouldn't overflow outside its boundaries. The decimal point shouldn't move, instead the value shouldn't be allowed to be increased over 100 to make sure it displays correctly.	

Feedback / Variable Updating

[Browse files](#)

I have now added the feature to edit game variables such as rocket angle and % thrust to control the rocket.

The data now comes in from the client on a separate namespace (websocket channel) to avoid confusion and mixed signals between the updating namespace and the standard namespace. This also means I don't have to parse for information from a very long string, and the messages to update are only received when required, rather than every ping meaning the program is more efficient.

In the ui.html I have added JS functions to utilise the numeric value changer I wrote. I've written functions that can be used with any ui method of changing data. They are completely generic and can be reused to change any values as long as those values are received by the game. I have written it as generic functions to avoid confusing and messy code, leaving it easy to read, as well as more efficient when adding in more functionality at a later date

by master

 meowterspace committed 14 days ago

1 parent 08e3898 commit b11944bb86102016b875df2634e6cb626fb8df7b

Showing 3 changed files with 247 additions and 25 deletions.

[Unified](#) [Split](#)

222  client/ui.html

```
23 23
24 24
25 25
26 +         width: 400px;
27 +         height: 600px;
28 +
29 +     }
30 +
31 +     #win-thm {
32 +         width: 150px;
33 +         height: 150px;
34 +         text-align: center;
35 +
36     #win-ang {
37         width: 165px;
38         height: 310px;
39         text-align: center;
40
41     #win-FDAI {
42         width: 400px;
43         height: 400px;
44         height: 100%;
45
46     }
47
48     td{
49         text-align: center;
50
51     }
52
53     input{
54         text-align: left;
55
56     }
57
58
59     #up{
60         width: 20px;
61         height: 20px;
62         background-color: #ffffdd0;
63
64     }
65
66     </style>
67
68     <script>
69
70
71
72     var latest_data = null;
73
74 +
75 +
```

```
76 +         // ===== EDIT STUFF =====
77 +         var update = io.connect('http://127.0.0.1:5000/update');
78 +
79 +
80 +         function make(id, value, dp){ // List of IDs for values, value, decimal point position in array
81 +             value = Math.round(value*100)/100;
82 +             console.log(value);
83 +             value = value.toString();
84 +             value = value.split('');
85 +             if (value.length < id.length){value.unshift('0');};
86 +             if (value[dp] == null){ value[dp] = '.'; };
87 +             for (var i=0; i<id.length; i++){
88 +                 if (value[i] == null){ value[i] = '0'; };
89 +
90 +                 document.getElementById(id[i]).innerHTML = value[i];
91 +             };
92 +         };
93 +
94 +
95 +
96 +         function change(type, power, direction){
97 +             console.log(value, power, direction);
98 +             switch(type){
99 +                 case 'thm':
100 +                     var value = latest_data.p_thm;
101 +                     value = value + (1*power*direction);
102 +                     make(['thm1', 'thm2', 'thm3', 'thm4'], value, 2);
103 +                     latest_data.p_thm = Math.round(value*100)/100;
104 +                     break;
105 +                 case 'angy':
106 +                     var value = latest_data.p_ang[0];
107 +                     value = value + (1*power*direction);
108 +                     make(['angy1', 'angy2', 'angy3', 'angy4', 'angy5', 'angy6'], value, 3);
109 +                     latest_data.p_ang[0] = Math.round(value*100)/100;
110 +                     break;
111 +                 case 'angp':
112 +                     var value = latest_data.p_ang[1];
113 +                     value = value + (1*power*direction);
114 +                     make(['angp1', 'angp2', 'angp3', 'angp4', 'angp5', 'angp6'], value, 3);
115 +                     latest_data.p_ang[1] = Math.round(value*100)/100;
116 +                     break;
117 +                 case 'angr':
118 +                     var value = latest_data.p_ang[2];
119 +                     value = value + (1*power*direction);
120 +                     make(['angr1', 'angr2', 'angr3', 'angr4', 'angr5', 'angr6'], value, 3);
121 +                     latest_data.p_ang[2] = Math.round(value*100)/100;
122 +                     break;
123 +             };
124 +             // UPDATE PAGE DATA HERE
125 +
126 +             update.send(latest_data);
127 +         };
128 +
129 +
130
131         /* This makes #winx draggable only by handle(class .handle)
132         with a change in opacity on event and ensures most recently dragged
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
```

```

107    188         //document.getElementById('time').innerHTML = 'Time ('+msg.zone+'): '+msg.time;
108    189         var readout;
109    200         cube.rotation.x = msg.angle[0];
110    210     };
111    211     socket.send(msg.uuid);
112    212     console.log(msg);
113    213   });
114    214   });
115    215 });
116    216   });
117    217   });
118    218   });
119    219   });
120    220   });
121    221   });
122    222   });
123    223   });
124    224   });
125    225   });
126    226   });
127    227   });
128    228   });
129    229   });
130    230   });
131    231   });
132    232   });
133    233   });
134    234   });
135    235   });
136    236   });
137    237   });
138    238   });
139    239   });
140    240   });
141    241   });
142    242   });
143    243   });
144    244   });
145    245   });
146    246   });
147    247   });
148    248   });
149    249   });
150    250   });
151    251   });
152    252   });
153    253   });
154    254   });
155    255   });
156    256   +<!-- ===== THM ===== -->
157    257   +<div id="win-thm" class="col-3 panel panel-default">
158    258     +<div class="handle panel-heading">Thrust Multiplier
159    259       +<button type="button" class="close" aria-label="Close" onclick="hideModal('win-thm')">
160    260         +<span aria-hidden="true">&times;</span>
161    261       +</button>
162    262     +</div>
163    263     +<div class="panel-body">
164    264       +<table>
165    265         +<tr>
166    266           +<td><input type='button' onclick="change('thm', 10, 1)" id='up' value='▲'></td>
167    267           +<td><input type='button' onclick="change('thm', 1, 1)" id='up' value='▲'></td>
168    268           +<td><input type='button' id='up'></td>
169    269           +<td><input type='button' onclick="change('thm', 0.1, 1)" id='up' value='▲'></td>
170    270         +</tr>
171    271         +<tr id='text'>
172    272           +<td id='thm1'>0</td>
173    273           +<td id='thm2'>0</td>
174    274           +<td id='thm3'>.</td>
175    275           +<td id='thm4'>0</td>
176    276           +<td>%</td>
177    277         +</tr>
178    278         +<tr>
179    279           +<td><input type='button' onclick="change('thm', 10, -1)" id='up' value='▼'></td>
180    280           +<td><input type='button' onclick="change('thm', 1, -1)" id='up' value='▼'></td>
181    281           +<td><input type='button' id='up'></td>
182    282           +<td><input type='button' onclick="change('thm', 0.1, -1)" id='up' value='▼'></td>
183    283         +</tr>
184    284         +</table>
185    285       +</div>
186    286     +</div>
187    287   +</div>
188    288   +<div id="win-ang" class="col-3 panel panel-default">
189    289     +<div class="handle panel-heading">Rocket Angle
190    290       +<button type="button" class="close" aria-label="Close" onclick="hideModal('win-ang')">
191    291         +<span aria-hidden="true">&times;</span>
192    292       +</button>
193    293     +</div>
194    294     +<div class="panel-body">
195    295       +<table>
196    296         +<tr>
197    297           +<td><input type='button' onclick="change('angy', 100, 1)" id='up' value='▲'></td>
198    298           +<td><input type='button' onclick="change('angy', 10, 1)" id='up' value='▲'></td>
199    299         +</tr>
200    300         +<tr>
201    301           +<td><input type='button' onclick="change('angy', 100, -1)" id='up' value='▼'></td>
202    302           +<td><input type='button' onclick="change('angy', 10, -1)" id='up' value='▼'></td>

```

```

303 +
304 <td><input type='button' onclick="change('angy', 1, 1)" id='up' value='▲'></td>
305 <td><input type='button' id='up'></td>
306 <td><input type='button' onclick="change('angy', 0.1, 1)" id='up' value='▲'></td>
307 <td><input type='button' onclick="change('angy', 0.01, 1)" id='up' value='▲'></td>
308 </tr>
309 <tr id='text'>
310 <td id='angy1'>0</td>
311 <td id='angy2'>0</td>
312 <td id='angy3'>0</td>
313 <td id='angy4'>.</td>
314 <td id='angy5'>0</td>
315 <td id='angy6'>0</td>
316 <td>°</td>
317 </tr>
318 <tr>
319 <td><input type='button' onclick="change('angy', 100, -1)" id='up' value='▼'></td>
320 <td><input type='button' onclick="change('angy', 10, -1)" id='up' value='▼'></td>
321 <td><input type='button' onclick="change('angy', 1, -1)" id='up' value='▼'></td>
322 <td><input type='button' id='up'></td>
323 <td><input type='button' onclick="change('angy', 0.1, -1)" id='up' value='▼'></td>
324 <td><input type='button' onclick="change('angy', 0.01, -1)" id='up' value='▼'></td>
325 </tr>
326 </table>
327 Pitch:
328 <table>
329 <tr>
330 <td><input type='button' onclick="change('angp', 100, 1)" id='up' value='▲'></td>
331 <td><input type='button' onclick="change('angp', 10, 1)" id='up' value='▲'></td>
332 <td><input type='button' onclick="change('angp', 1, 1)" id='up' value='▲'></td>
333 <td><input type='button' id='up'></td>
334 <td><input type='button' onclick="change('angp', 0.1, 1)" id='up' value='▲'></td>
335 <td><input type='button' onclick="change('angp', 0.01, 1)" id='up' value='▲'></td>
336 </tr>
337 <tr id='text'>
338 <td id='angp1'>0</td>
339 <td id='angp2'>0</td>
340 <td id='angp3'>0</td>
341 <td id='angp4'>.</td>
342 <td id='angp5'>0</td>
343 <td id='angp6'>0</td>
344 <td>°</td>
345 </tr>
346 <tr>
347 <td><input type='button' onclick="change('angp', 100, -1)" id='up' value='▼'></td>
348 <td><input type='button' onclick="change('angp', 10, -1)" id='up' value='▼'></td>
349 <td><input type='button' onclick="change('angp', 1, -1)" id='up' value='▼'></td>
350 <td><input type='button' id='up'></td>
351 <td><input type='button' onclick="change('angp', 0.1, -1)" id='up' value='▼'></td>
352 <td><input type='button' onclick="change('angp', 0.01, -1)" id='up' value='▼'></td>
353 </tr>
354 </table>
355 Roll:
356 <table>
357 <tr>
358 <td><input type='button' onclick="change('angr', 100, 1)" id='up' value='▲'></td>
359 <td><input type='button' onclick="change('angr', 10, 1)" id='up' value='▲'></td>
360 <td><input type='button' onclick="change('angr', 1, 1)" id='up' value='▲'></td>
361 <td><input type='button' id='up'></td>
362 <td><input type='button' onclick="change('angr', 0.1, 1)" id='up' value='▲'></td>
363 <td><input type='button' onclick="change('angr', 0.01, 1)" id='up' value='▲'></td>
364 </tr>
365 <tr id='text'>
366 <td id='angr1'>0</td>
367 <td id='angr2'>0</td>
368 <td id='angr3'>0</td>
369 <td id='angr4'>.</td>
370 <td id='angr5'>0</td>
371 <td id='angr6'>0</td>
372 <td>°</td>
373 </tr>

```

```

373 +          <tr>
374 +          <td><input type='button' onclick="change('angr', 100, -1)" id='up' value='▼'></td>
375 +          <td><input type='button' onclick="change('angr', 10, -1)" id='up' value='▼'></td>
376 +          <td><input type='button' onclick="change('angr', 1, -1)" id='up' value='▼'></td>
377 +          <td><input type='button' id='up'></td>
378 +          <td><input type='button' onclick="change('angr', 0.1, -1)" id='up' value='▼'></td>
379 +          <td><input type='button' onclick="change('angr', 0.01, -1)" id='up' value='▼'></td>
380 +      </tr>
381 +    </table>
382 +  </div>
383 +</div>
173 384 </body>
385 +
386 +
387 +
388 +
389 +

```

13 ████████ server/app.py

```

182 182
183 183     @socketio.on('message')
184 184     def handle_message(message):
185 185         - print('setup')
186 185         for i in resources.OBJECTS:
187 186             if (i[1] == 'planet'):
188 187                 resources.run(i[0], player, 0.1)
189 188
190 190         - if message == str(meta['uuid']): print(message)
191 191         - else: print('Packet Loss!')
192 192         + #if message != str(meta['uuid']): #print(message)
193 193         + # print('Packet Loss!')
194 194
195 195         meta['time'] = str(datetime.datetime.now())
196 196         meta['serv'] = str(datetime.datetime.now()-start)
197 197         to_send.update(resources.data)
198 198         send(to_send)
199 199
200 200
201 201     -@socketio.on('message', namespace='/edit_data')
202 201     +@socketio.on('message', namespace='/update')
203 202     def handle_incoming_data(message):
204 203         - print('Incoming Data: '+message)
205 204         + send(resources.data)
206 205         + print('Incoming Data: '+str(message))
207 206         + resources.update(player, message)
208 207         + print(resources.data)
209 208         + print('Data updated')
210 209
211 210     @socketio.on('message', namespace='/lobbu')
212 210     def handle_lobby_message(message):

```

37 ████████ server/resources.py

```

88 88         self.length, self.F, self.angle = length, F, angle
89 89         self.q, self.Ve, self.Pe, self.Ae = q, Ve, Pe, Ae
90 90         self.Cd, self.Vg, self.fuel = Cd, Vg, Fuel
91 91         + self.thm = 0;
92 92         def resolve_thrust(self): # yaw, pitch, roll -> x, y, z
93 93             x = self.F*(np.cos(self.angle[0])*np.cos(self.angle[1]))
94 94             y = self.F*(np.sin(self.angle[1]))
138 139         def run(planet, player, response_t):
139 140             global OBJECTS
140 141             global data
141 142             - print(player.pos)
142 142             + #print(player.pos)
143 143             F = [0, 0, 0]
144 144             player.F = [0, 0, 0]
144 145             - player.F = player.thrust(planet)
145 145             + player.F = player.thrust(planet)*player.thm
146 146             F = player.resolve_thrust()

```

```

146 147     for i in OBJECTS:
147 148         if i[1] == 'planet':
158 159             s = [(u[0]*response_t) + (0.5*player.a[0]*(response_t**2)), (u[1]*response_t) + (0.5*player.a[1]*(response_t**2)),
160             player.pos = [player.pos[0]+s[0], player.pos[1]+s[1], player.pos[2]+s[2]]
160
161         - print('a = '+str(player.a))
161         - print('v = '+str(player.V))
162         - print('x = '+str(player.pos[0]))
163         - print('y = '+str(player.pos[1]))
164         - print('z = '+str(player.pos[2]))
161
162         + #print('F = '+str(player.F))
162         + #print('a = '+str(player.a))
163         + #print('v = '+str(player.V))
164         + #print('x = '+str(player.pos[0]))
165         + #print('y = '+str(player.pos[1]))
166         + #print('z = '+str(player.pos[2]))
165
166         data['p_acc'] = player.a
167         data['p_vel'] = player.V
168         data['p_pos'] = player.pos
169         data['p_ang'] = player.angle
170         data['p_fue'] = 100 # ADD THIS
171         - data['p_thm'] = 100 # ADD THIS
173         + data['p_thm'] = player.thm # ADD THIS
172         data['p_sta'] = True # ADD THIS
173         data['p_orb'] = False # ADD THIS
174
175
176     -def update_data(data):
178     +def update(player, data):
179         global OBJECTS
180         player.a = data['p_acc']
181         player.V = data['p_vel']
180         player.pos = data['p_pos']
181         - player.angle = data['p_pos']
183         + player.angle = data['p_ang']
184         + player.thm = data['p_thm']
182
183         #FUE, THM, STA, ORB
184
184
194     # STA Grav 2
195     # STA
196
197     -Player = setup()
198     -print('ok')
199     -for object in OBJECTS:
200         - if object[1] == 'planet':
201             - for i in range(100):
202                 - print(i)
203                 - run(object[0], Player, 1)
200
201     +#Player = setup()
201     +#print('ok')
202     +#for object in OBJECTS:
203     +#     if object[1] == 'planet':
204     +#         for i in range(100):
205     +#             print(i)
206     +#             run(object[0], Player, 1)
204
207     0

```

1 comment on commit b11944b



meowterspace commented on b11944b 10 minutes ago

Owner

Preview

Thrust Multiplier

Current Thrust:

0 1 1 8 %

Rocket Angle

Yaw:

1 0 7 . 2 3 °

Pitch:

0 0 0 . 0 0 °

Roll:

0 0 0 . 0 0 °

readout

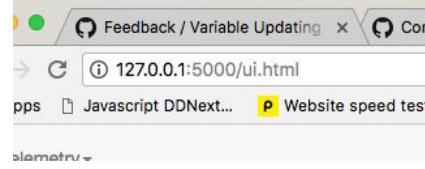
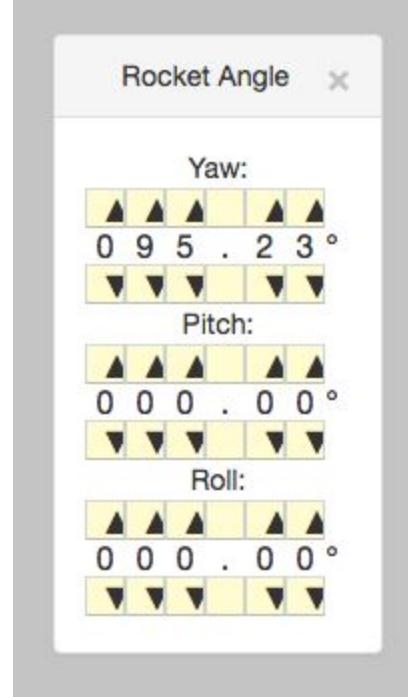
```

undefined
p_acc: 111181.55517242692,0,49098.201488858926
p_ang: 107.23,0,0
p_fue: 100
p_orb: false
p_pos: 6e+24,0,5486544536.996351
p_sta: true
p_thm: 118
p_vel: 18448639.875734493,0,13978796.51252677
serv: 0:27:33.152431
time: 2018-04-11 22:57:07.716080
uuid: 7afe90c5-fa17-49b9-8ba4-f6fa0b3c1eff
zone: GMT

```

p_acc: 111181.55517242692,0,49098.201488858926
p_ang: 107.23,0,0
p_fue: 100
p_orb: false
p_pos: 6e+24,0,5486544536.996351
p_sta: true
p_thm: 118
p_vel: 18448639.875734493,0,13978796.51252677
serv: 0:27:33.152431
time: 2018-04-11 22:57:07.716080
uuid: 7afe90c5-fa17-49b9-8ba4-f6fa0b3c1eff
zone: GMT

incoming Data: {"time": "2018-04-11 22:30:31.078366", "zone": "GMT", "serv": "0:00:56.514718", "uuid": "369246f2-c704-41d6-b6af-8207d6f5f24e", "p_ang": [95.1, 0, 0], "p_fue": 100, "p_thm": 6, "p_sta": True, "p_orb": False}
Data updated
incoming Data: {"time": "2018-04-11 22:30:31.078366", "zone": "GMT", "serv": "0:00:56.514718", "uuid": "369246f2-c704-41d6-b6af-8207d6f5f24e", "p_ang": [4512.472606947986, 0, 4222.557397069228], "p_fue": [-1372195.2781759469, 0, 2688533.0874360804], "p_pos": [6e+24, 0, 579018137.2794912], "p_ang": [95.1, 0, 0], "p_fue": 100, "p_thm": 6, "p_sta": True, "p_orb": False}
{"p_acc": [4512.472606947985, 0, 4222.5573970692276], "p_fue": [-1371744.022915252, 0.0, 2688955.2631757874], "p_pos": [5.999999999999999e+24, 0.0, 579287011.69302177], "p_ang": [95, 0, 0], "p_fue": 100, "p_thm": 6, "p_sta": True, "p_orb": False}
Data updated
incoming Data: {"time": "2018-04-11 22:30:31.212347", "zone": "GMT", "serv": "0:00:56.648699", "uuid": "c7d7bbdb-bd4d-4aba-b0b5-b1e808667e0a", "p_ang": [4068.376707679505, 0, 4651.9577560862335], "p_fue": [-1366092.785114282, 0, 2695510.9440702084], "p_pos": [6e+24, 0, 583056170.2431207], "p_ang": [95.2, 0, 0], "p_fue": 100, "p_thm": 6, "p_sta": True, "p_orb": False}
{"p_acc": [4068.376707679502, 0, 4651.9577560862335], "p_fue": [-1365685.8674436683, 0.0, 2695976.1398458169], "p_pos": [5.999999999999999e+24, 0.0, 579332744.5973165], "p_ang": [95.1, 0, 0], "p_fue": 100, "p_thm": 6, "p_sta": True, "p_orb": False}
Data updated
incoming Data: {"time": "2018-04-11 22:30:31.692279", "zone": "GMT", "serv": "0:00:57.038624", "uuid": "86f35f05-df42-405d-8b81-1f1c59554367", "p_ang": [3583.630933174366, 0, 5034.877290937271], "p_fue": [-1350683.0921017763, 0, 2717160.91642123], "p_pos": [6e+24, 0, 594693414.7431772], "p_ang": [95.21, 0, 0], "p_fue": 100, "p_thm": 6, "p_sta": True, "p_orb": False}
{"p_acc": [3583.630933174366, 0, 5034.877290937271], "p_fue": [-1350324.7290084588, 0.0, 2717664.4041503235], "p_pos": [5.999999999999999e+24, 0.0, 594965156.00920571], "p_ang": [95.2, 0, 0], "p_fue": 100, "p_thm": 6, "p_sta": True, "p_orb": False}
Data updated
incoming Data: {"time": "2018-04-11 22:30:31.730559", "zone": "GMT", "serv": "0:00:57.166909", "uuid": "40903088-c86f-4ad9-89fa-343fb63ce594", "p_ang": [3533.103819353573, 0, 5070.461261233458], "p_fue": [-1346090.0571366155, 0, 2723752.5160608357], "p_pos": [6e+24, 0, 598230008.47429], "p_ang": [95.22, 0, 0], "p_fue": 100, "p_thm": 6, "p_sta": True, "p_orb": False}
{"p_acc": [3533.103819353573, 0, 5070.461261233457], "p_fue": [-1345736.7467546801, 0.0, 2724259.5621869592], "p_pos": [5.999999999999999e+24, 0.0, 598502409.07828237], "p_ang": [95.21, 0, 0], "p_fue": 100, "p_thm": 6, "p_sta": True, "p_orb": False}
Data updated
incoming Data: {"time": "2018-04-11 22:30:31.830207", "zone": "GMT", "serv": "0:00:57.266557", "uuid": "ff83bc83-9ab1-4a5f-afcd-458d0bab853", "p_ang": [3482.223398085819, 0, 5105.538189620943], "p_fue": [-1342687.8337385214, 0, 2728858.0542584657], "p_pos": [6e+24, 0, 600956313.7594452], "p_ang": [95.23, 0, 0], "p_fue": 100, "p_thm": 6, "p_sta": True, "p_orb": False}
{"p_acc": [3482.223398085819, 0, 5105.538189620942], "p_fue": [-1342259.611398712, 0.0, 2729368.6080694287], "p_pos": [5.999999999999999e+24, 0.0, 601229225.09256131], "p_ang": [95.22, 0, 0], "p_fue": 100, "p_thm": 6, "p_sta": True, "p_orb": False}
Data updated

Test	Pass/Fail	Description	Evidence
Check that a flask server can be started	PASS	Look at the server terminal. The server should quote a debugger pin and a server address. Important to check the server works before continuing.	<pre>JSONIFY_MIMETYPE application/json TEMPLATES_AUTO_RELOAD None * Debugger is active! * Debugger PIN: 132-511-971 (20710) wsgi starting up on http://127.0.0.1:5000 (20710) accepted ('127.0.0.1', 59344) (20710) accepted ('127.0.0.1', 59345) 127.0.0.1 - - [11/Apr/2018 22:29:38] "GET /ui.html HTTP/1.1 127.0.0.1 - - [11/Apr/2018 22:29:38] "GET /resources/js/thr 127.0.0.1 - - [11/Apr/2018 22:29:38] "GET /resources/js/nav 127.0.0.1 - - [11/Apr/2018 22:29:38] "GET /socket.io/?EIO=3</pre>
Check that the UI page loads	PASS	Page should load without error from a server. This is important to check after every change to make sure the page is unaffected. The JS error log should be empty	
Check that numerical variable editors for angle changes server data	PASS	Observing the server readout in /ui.html should show that the value (pulled straight from server over websockets) changes in accordance with the NVE value. This is to check that the data can be sent from client and directly edit angle variables in the computational engine (resources.py)	

[Browse files](#)

Utilised Multi-Threading

The computations are now run in a completely separate thread. This is because the previous solution was flawed. Previously the computation was run every time a request was made by the client to the server and the server would calculate a ping time and use that as the time interval between calculations however I would later encounter the problem that in a multiplayer scenario there would be multiple pings to the server and the the calculations would be run multiple times with the same time parameter during the time. So the calculations could be run 4 times (0.4 seconds worth) during 0.1 seconds etc.

To fix this the calculations now runs in a separate thread to the server, and the server fetches the results from the resources script rather than the run() function returning data

by master

 meowterspace committed 12 days ago

1 parent b11944b commit 97f4b7136938394e06c348736e86251ba1d0474d

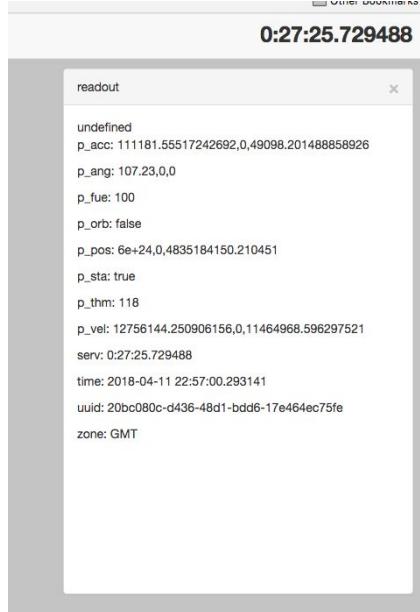
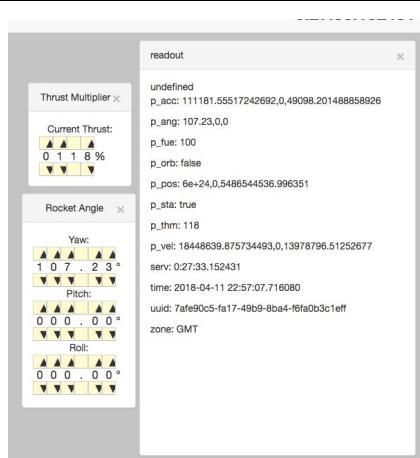
Showing 1 changed file with 12 additions and 6 deletions.

[Unified](#) [Split](#)

18  server/app.py

```
... ... @@ -1,6 +1,7 @@
1 1 import json
2 2 import uuid
3 3 import datetime
4 4 +import time
5 5 import os
6 6 import io
7 7 import errno
182 183
183 184 @socketio.on('message')
184 185 def handle_message(message):
185 - for i in resources.OBJECTS:
186 - if (i[1] == 'planet'):
187 -     resources.run(i[0], player, 0.1)
188 186
189 - #if message != str(meta['uuid']): #print(message)
190 - # print('Packet Loss!')
191 187
192 188     meta['time'] = str(datetime.datetime.now())
193 189     meta['serv'] = str(datetime.datetime.now()-start)
200 196
201 197 @socketio.on('message', namespace='/update')
202 198 def handle_incoming_data(message):
203 - send(resources.data)
204 199     print('Incoming Data: '+str(message))
205 200     resources.update(player, message)
206 201     print(resources.data)
210 205 def handle_lobby_message(message):
211 206     send(GAME)
212 207
208 +===== THREADS =====
209 +
210 +def compute(time):
211 +    while True:
212 +        for i in resources.OBJECTS:
213 +            if (i[1] == 'planet'): resources.run(i[0], player, time)
214 +            time.sleep(time)
215 +
216 +
217 +    if __name__ == '__main__':
218 +        compute_thread = Thread(target=compute, args=(0.1))
219 +        compute_thread.start()
220         socketio.run(app)
```

030418M1

Test	Pass/Fail	Description	Evidence																																																												
Check that using multithreading has no effect on the server starting	PASS	The server should start correctly with an IP and debugger PIN to show that it still works regardless of the multithreading setup	 <pre>JSON_SORT_KEYS True JSONIFY_PRETTYPRINT_REGULAR True JSONIFY_MIMETYPE application/json TEMPLATES_AUTO_RELOAD None * Debugger is active! * Debugger PIN: 132-511-971 (21745) wsgi starting up on http://127.0.0.1:5000</pre>																																																												
Check that multithreading has no effect on the data the client receives	PASS	The server readout in /ui.html should be full and every value should have a value instead of an empty string or error to show that data can still be sent between the two threads (server and computational)	 <p>0:27:25.729488</p> <p>readout</p> <pre>undefined p_acc: 111181.55517242692,0,49098.201488858926 p_ang: 107.23,0 p_fue: 100 p_orb: false p_pos: 6e+24,0,4835184150.210451 p_sta: true p_thm: 118 p_vel: 12756144.250906156,0,11464968.596297521 serv: 0:27:25.729488 time: 2018-04-11 22:57:00.293141 uuid: 20bc080c-d436-48d1-bdd6-17e464ec75fe zone: GMT</pre>																																																												
Check that multithreading has no effect on the client's ability to change server data	PASS	The server readout in /ui.html should still reflect the value on the NVE to make sure data can pass between threads fine in the server without any adverse effects.	 <p>readout</p> <p>Thrust Multiplier</p> <p>Current Thrust:</p> <table border="1"> <tr><td>▲</td><td>▲</td><td>▲</td><td>▲</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>8 %</td></tr> <tr><td>▼</td><td>▼</td><td>▼</td><td>▼</td></tr> </table> <p>Rocket Angle</p> <p>Yaw:</p> <table border="1"> <tr><td>▲</td><td>▲</td><td>7</td><td>2</td><td>3 °</td></tr> <tr><td>1</td><td>0</td><td>7</td><td>.</td><td>2</td><td>3</td></tr> <tr><td>▼</td><td>▼</td><td>▼</td><td>▼</td><td>▼</td></tr> </table> <p>Pitch:</p> <table border="1"> <tr><td>▲</td><td>▲</td><td>0</td><td>0</td><td>0 °</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>.</td><td>0</td><td>0</td></tr> <tr><td>▼</td><td>▼</td><td>▼</td><td>▼</td><td>▼</td></tr> </table> <p>Roll:</p> <table border="1"> <tr><td>▲</td><td>▲</td><td>0</td><td>0</td><td>0 °</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>.</td><td>0</td><td>0</td></tr> <tr><td>▼</td><td>▼</td><td>▼</td><td>▼</td><td>▼</td></tr> </table> <pre>undefined p_acc: 111181.55517242692,0,49098.201488858926 p_ang: 107.23,0 p_fue: 100 p_orb: false p_pos: 6e+24,0,5486544536.996351 p_sta: true p_thm: 118 p_vel: 18448639.875734493,0,13978796.51252677 serv: 0:27:33.152431 time: 2018-04-11 22:57:07.716080 uuid: 7afe90c5-fa17-49b9-8ba4-46fa0b3c1eff zone: GMT</pre>	▲	▲	▲	▲	0	1	1	8 %	▼	▼	▼	▼	▲	▲	7	2	3 °	1	0	7	.	2	3	▼	▼	▼	▼	▼	▲	▲	0	0	0 °	0	0	0	.	0	0	▼	▼	▼	▼	▼	▲	▲	0	0	0 °	0	0	0	.	0	0	▼	▼	▼	▼	▼
▲	▲	▲	▲																																																												
0	1	1	8 %																																																												
▼	▼	▼	▼																																																												
▲	▲	7	2	3 °																																																											
1	0	7	.	2	3																																																										
▼	▼	▼	▼	▼																																																											
▲	▲	0	0	0 °																																																											
0	0	0	.	0	0																																																										
▼	▼	▼	▼	▼																																																											
▲	▲	0	0	0 °																																																											
0	0	0	.	0	0																																																										
▼	▼	▼	▼	▼																																																											

integrated orbit check

[Browse files](#)

As seen in previous experiment, I've added a check to see if the rocket is in an orbit. This will add an objective to the game that the player can aim for

by master

 meowterspace committed 7 days ago

1 parent 97f4b71 commit 834c7456fcf21e13e406d5d27295df62825034ee

Showing 1 changed file with 8 additions and 4 deletions.

[Unified](#) [Split](#)

12  server/resources.py

```
101 101
102 102         return F # Ae = area of exit
103 103
104 104     - def drag_area():
105 105     -     return 1
106 106     + def drag_area(v, a, r, h):
107 107     +     area = r*h
108 108     +     top = np.arccos((V[0]*a[0]+V[1]*a[1]+V[2]*a[2]))
109 109     +     bottom = ((np.sqrt((a[0]**2)+(a[1]**2)+(a[2]**2)))*np.sqrt((V[0]**2)+(V[1]**2)+(V[2]**2)))
110 110     +     area = area * (top/bottom)
111 111     +     return area
112 112
113 113     def drag(planet, T, h): # P = Pressure /PA, T = temp(k), Vg = velocity
114 114     -     F = 0.5 * ((planet.pressure(T, h) / (286 * T)) * (self.V ** 2) * self.Cd * self.drag_area()) # Cd = Coefficient
115 115     +     F = 0.5 * ((planet.pressure(T, h) / (286 * T)) * (self.V ** 2) * self.Cd * self.drag_area(self.V, self.a, self.
116 116     +     return F
117 117
118 118     def die():
119 119
120 120     def make_planet(Planet, name, mass, radius, pos, V, a, p0, molMass):
121 121     -     exec(str(name)+' = Planet('+str(mass)+','+str(radius)+','+str(pos)+','+str(V)+','+str(a)+','+str(p0)+','+str(molMass)+'
122 122     +     exec('OBJECTS.append( ['+str(name)+', "planet"] )')
123 123
124 124
125 125
126 126     -     +
127 127     # funct:
128 128     def setup():
```

2 comments on commit 834c745

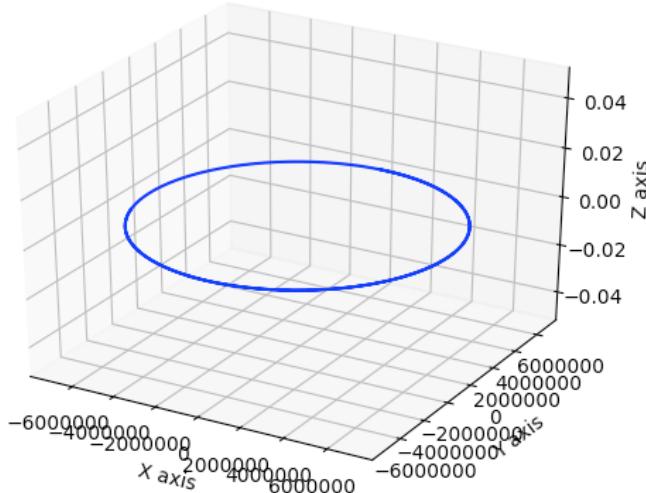


meowterspace replied 13 minutes ago

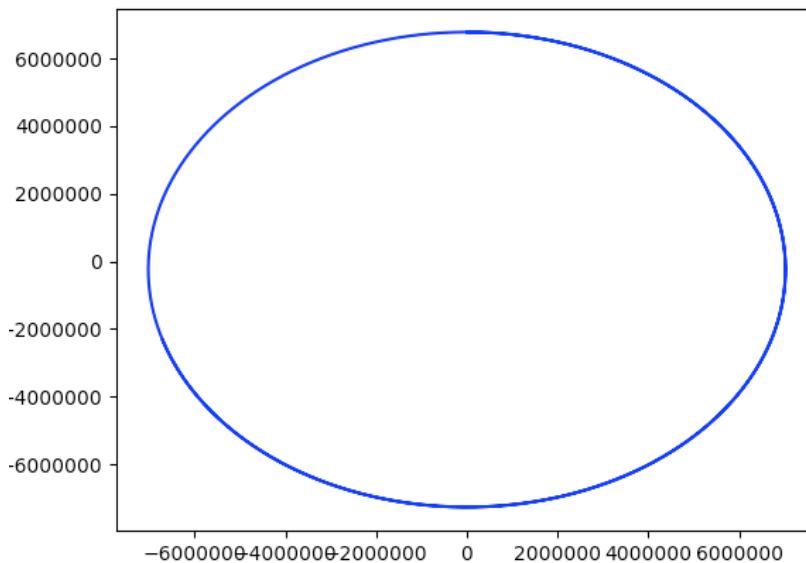
[Owner](#)

Preview

Correct orbit:



```
Cat$ iMac:server Cat$ python3 /Users/Ben/Desktop/CS\ Testing/150318E1/mission_cont
026de27364de6460cc1bab436dc38f2/Computation\ Engine/Compute_engine_test.py
{{a = list}}
{{list}}
[[ 7.8000000e+02   1.5599999e+03   2.3399999e+03 ..., -6.70074255e+06
-6.70096446e+06 -6.70118629e+06]
 [ 6.77899991e+06   6.77899974e+06   6.77899948e+06 ..., -2.33618329e+06
-2.33547155e+06 -2.33475978e+06]
 [ 0.0000000e+00   0.0000000e+00   0.0000000e+00 ...,  0.0000000e+00
 0.0000000e+00   0.0000000e+00]]
```



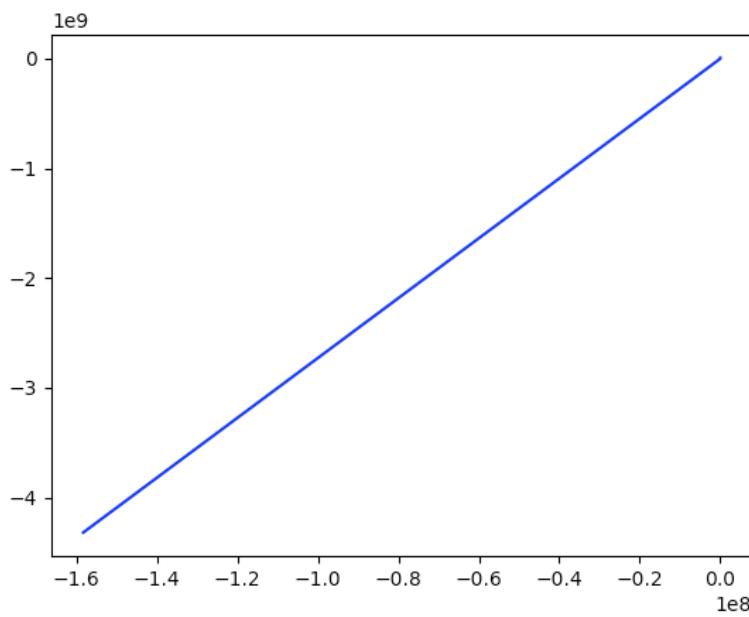
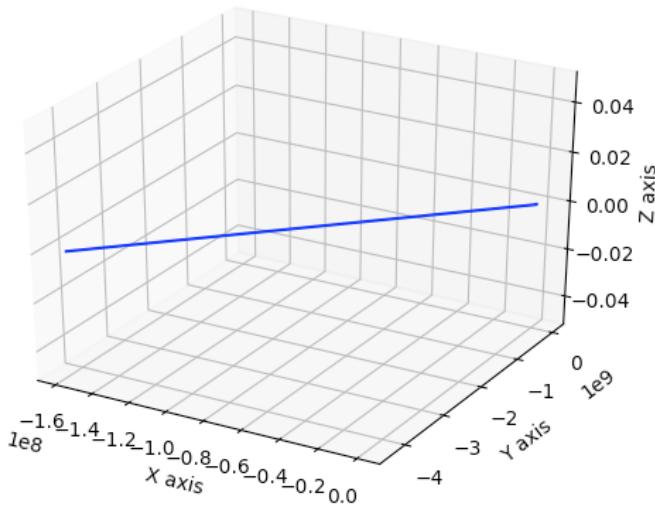
```
7096266.69238
__orbit__
1884185.89591
True
Cat$ iMac:server
```

Function returns True

meowterspace replied 13 minutes ago

Owner

Unsuccessful orbit:

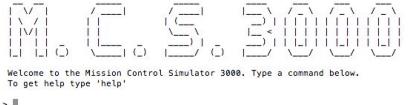
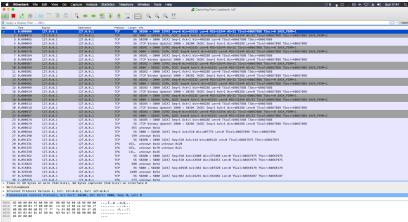


```

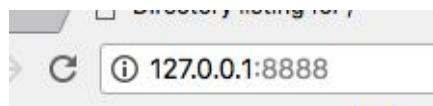
{{{a = list}}}
{{{list}}}
-----
[[ 1.0000000e+00  1.9999999e+00  2.9999995e+00 ..., -1.58282301e+08
-1.58284056e+08 -1.58285811e+08]
[ 6.77899991e+06  6.77899974e+06  6.77899948e+06 ..., -4.32023267e+09
-4.32028058e+09 -4.32032849e+09]
[ 0.00000000e+00  0.00000000e+00  0.00000000e+00 ...,  0.00000000e+00
 0.00000000e+00  0.00000000e+00]]
-----
432327107.64
__orbit__
-3.38318558928e+15
False
Cats-iMac:server Cat
Orbit returns False

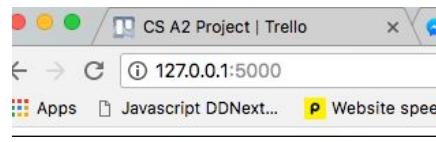
```

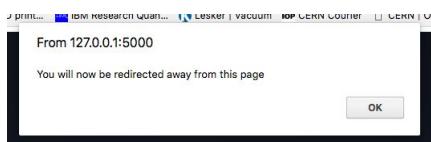
4.0 Final Testing

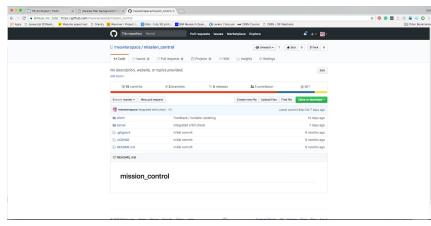
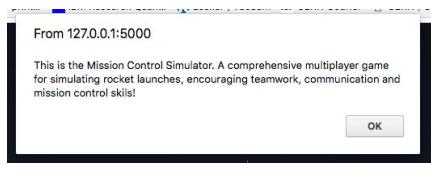
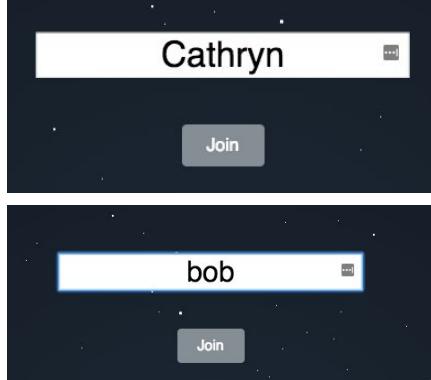
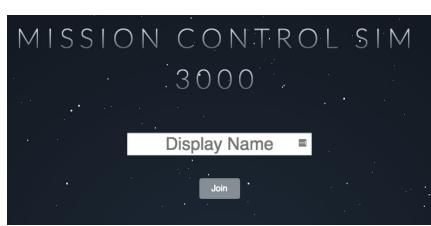
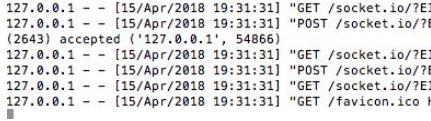
No.	Description	Pass/Fail	Expected output	Actual output
1	Admin menu	PASS	Admin menu should open without error. This is to check that there are no errors in the python code at the end of development.	<pre>Cats-Mac:~/server/Cats python3 run.py</pre> 
2	Admin title	PASS	Text 'M.C.S 3000' should appear in ascii art. This proves that the code actually runs and gives the admin user a nice introduction to the program.	
3	Admin help	PASS	A help menu should appear with a list of different functions you can type into the terminal to perform different functions. It's important to check this to make sure the user can get the help they need to operate the program.	<pre>> help()</pre> <pre>start : starts server stop : shuts down server halt : stops the server instantly help : displays list of commands exit : closes this window, not the server</pre>
4	Server start	PASS	The terminal should print the CONFIG.txt options and end with a debugger pin and the server address. This test shows that the subprocess has started correctly and that the server can still boot correctly in this new version of the code. If you check the LAN in wireshark you should see a server on the server IP and port.	

				<pre> ['app.pyw', '127.0.0.1', '5000'] DEBUG False TESTING False PROPAGATE_EXCEPTIONS None PRESERVE_CONTEXT_ON_EXCEPTION None SECRET_KEY secret PERMANENT_SESSION_LIFETIME 31 days, 0:00:00 USE_X_SENDFILE False LOGGER_NAME __main__ LOGGER_HANDLER_POLICY always SERVER_NAME None APPLICATION_ROOT None SESSION_COOKIE_NAME session SESSION_COOKIE_DOMAIN None SESSION_COOKIE_PATH None SESSION_COOKIE_HTTPONLY True SESSION_COOKIE_SECURE False SESSION_REFRESH_EACH_REQUEST True MAX_CONTENT_LENGTH None SEND_FILE_MAX_AGE_DEFAULT 12:00:00 TRAP_BAD_REQUEST_ERRORS False TRAP_HTTP_EXCEPTIONS False EXPLAIN_TEMPLATE_LOADING False PREFERRED_URL_SCHEME http JSON_AS_ASCII True JSON_SORT_KEYS True JSONIFY_PRETTYPRINT_REGULAR True JSONIFY_MIMETYPE application/json TEMPLATES_AUTO_RELOAD None * Debugger is active! </pre>
5	Server stop command	PASS	The terminal should verify that you want to kill the python process. This check is required to make sure the user has an option to abort the stop procedure since it will be killing all python services.	<pre> > stop() WARNING: This will kill all running python processes on your computer. Do you wish to continue? Y/N : Y HALT Killed: 9 Cats-iMac:server Cats </pre>
6	Server stop	PASS	The terminal session should close along with the server. Use wireshark to verify that no server is running on the network. This check ensures that the server can be shut down and that the sub-process can be killed using the stop() command.	<pre> > stop() WARNING: This will kill all running python processes on your computer. Do you wish to continue? Y/N : Y HALT Killed: 9 Cats-iMac:server Cats </pre>
7	Server abort stop	PASS	The terminal window should not close and the server should continue running as normal. This step is important to make sure the stop() function options work correctly	<pre> > stop() WARNING: This will kill all running python processes on your computer. Do you wish to continue? Y/N : Y </pre>

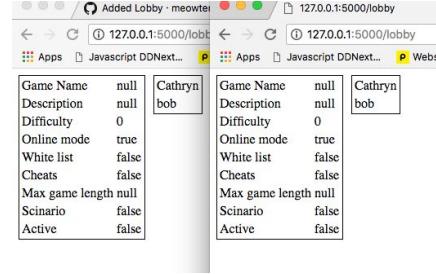
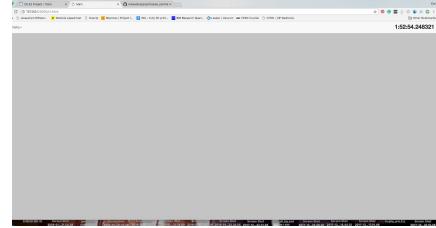
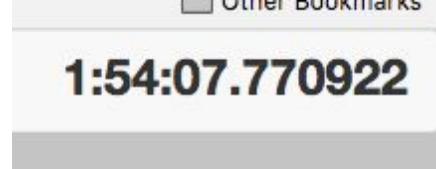
			and don't kill all python processes unnecessary..	
8	Server invalid input stop	PASS	The terminal should ask you to type Y or N again. This is important to give the user feedback and not to take any input. The check is to make sure that it doesn't continue with any other processes.	<pre>> stop() WARNING: This will kill all running python processes Do you wish to continue? Y/N : yes Do you wish to continue? Y/N : no Do you wish to continue? Y/N : y Do you wish to continue? Y/N : n Do you wish to continue? Y/N : k Do you wish to continue? Y/N : abc Do you wish to continue? Y/N : 0 Do you wish to continue? Y/N : </pre>
9	Server IP Change	PASS	The start server text should display the new IP instead of 127.0.0.1, and if you navigate to the new IP the server should be running there. You can verify this with wireshark. This test is important because some users don't want to run on their local machine and may wish to open the game up to WAN not LAN so the IP change feature must be tested.	<pre>* Debugger PIN: 132-511-971 wsgi starting up on http://192.168.10.142:8888</pre>
10	Server Port Change	PASS	The start server text should display the new port instead of 127.0.0.1, and if you navigate to the new IP the server should be running there. You can verify this with wireshark. This test is important because some users don't want to run on their local machine and may wish to open the game up to WAN not LAN so the port change feature must be tested.	 <pre>* Debugger PIN: 132-511-971 wsgi starting up on http://192.168.10.142:8888</pre>
11	Server IP Change on	PASS	The start server text should display	

	start		<p>192.168.10.142 instead of 127.0.0.1, and if you navigate to the new IP the server should be running there. You can verify this with wireshark. This test is important because some users don't want to run on their local machine and may wish to open the game up to WAN not LAN so the IP change feature must be tested.</p>	
12	Server Port Change on start	PASS	<p>The start server text should display the new port 888 instead of 5000, and if you navigate to the new IP the server should be running there. You can verify this with wireshark. This test is important because some users don't want to run on their local machine and may wish to open the game up to WAN not LAN so the port change feature must be tested.</p>	
13	User connect to server index	PASS	<p>While the server is running the index page should load and the server terminal should display a GET request. This check makes sure that the server is accessible at the correct address via browser.</p>	<pre>127.0.0.1 - - [15/Apr/2018:17:47:07] "GET / HTTP/1.1" 200 43937 0.001690 127.0.0.1 - - [15/Apr/2018:17:47:19] "GET /resources/19/anykey.js HTTP/1.1" 200 2353 0.003193 127.0.0.1 - - [15/Apr/2018:17:47:19] "GET /favicon.ico HTTP/1.1" 200 176 0.004625 [2643] accepted ('127.0.0.1', 51899)</pre>
14	Check that an index page loads	PASS	<p>The index page should be served by the server and the browser should show the page loading</p>	

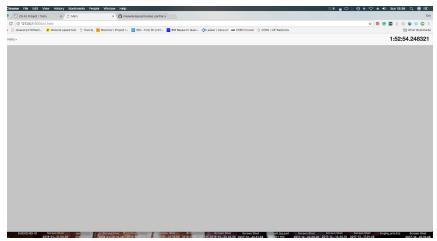
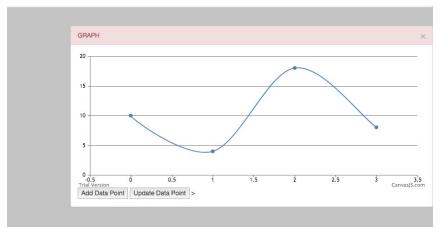
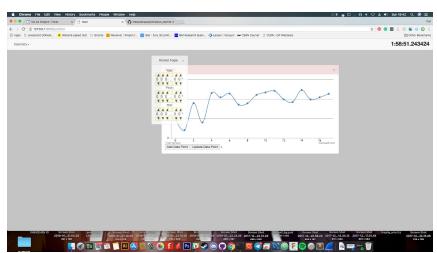
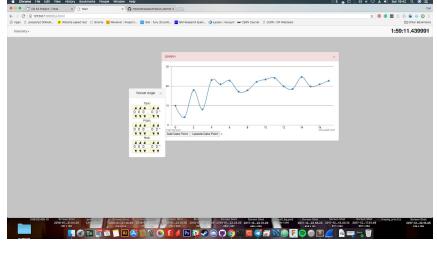
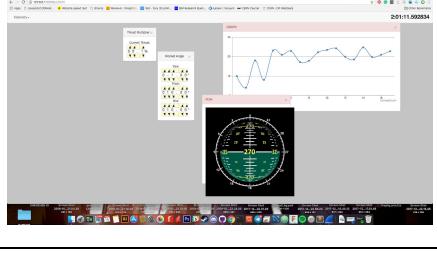
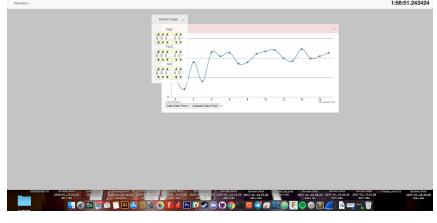
			with no critical errors in the JS console or log. This check is needed to make sure there are no page errors in the final version	
15	Check that the index page renders the defined index page	PASS	The index page served should be index.html. This makes sure that Flask is rendering the correct page and the app route is correct.	
16	Check that the index page has a functioning press any key to continue feature	PASS	Each key should trigger the join/host/about menu to appear. This makes sure that there isn't a key that you can press that won't trigger the menu and that the ANY part of press any key holds true.	
17	Check join button functions	PASS	The menu should disappear and show a text-field. This is to check that the javascript functions are called correctly and that this button functions as needed in the final product.	
18	Check host button functions	PASS	The page should display an alert saying you will be redirected to a different page, and then the project github page should load. This makes sure that the host button works as needed and the page	

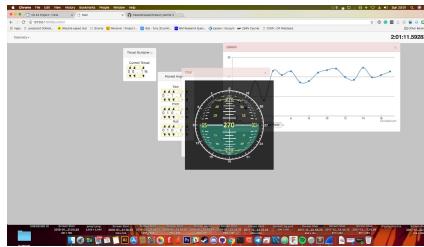
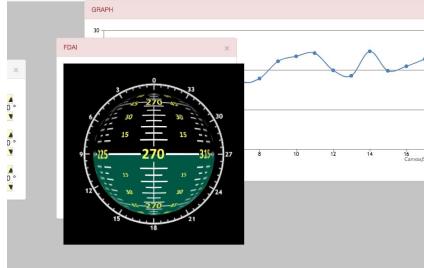
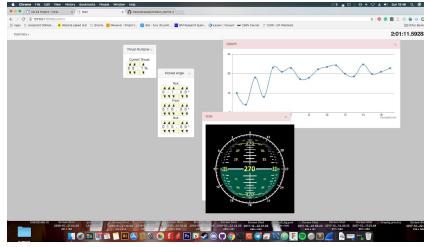
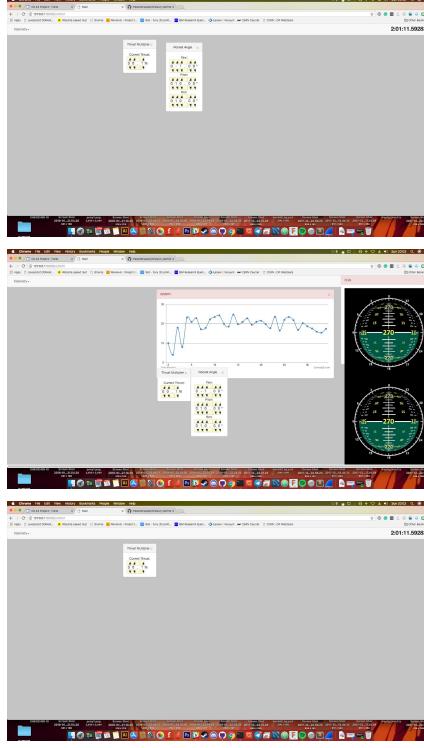
			redirects successfully.	
19	Check about button functions	PASS	The page should display an alert giving some about text. This test is needed to make sure the about button works so confused users can get more information.	
20	Check name text-field works	PASS	The characters should appear in the text box as you type them. This makes sure that the textbox can display ASCII correctly.	
21	Check name text-field can't be empty	Fail	Nothing should happen. This makes sure the javascript check works and the user can't submit a blank name to the server. I need to test this to ensure this key validity check works.	
22	Check the final join button redirects to lobby	PASS	The page should be directed to /lobby. This check is needed to make sure the page still directs in the final version.	
23	Check /lobby loads correctly	PASS	The page should load with zero errors to check that the page redirect is correct and accepted by	

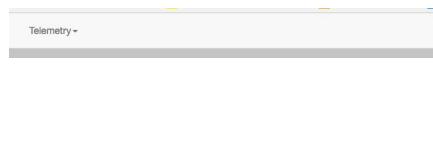
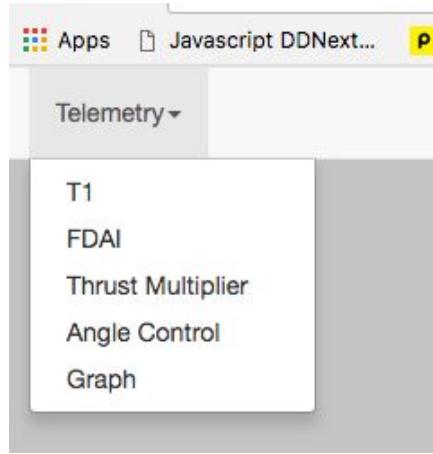
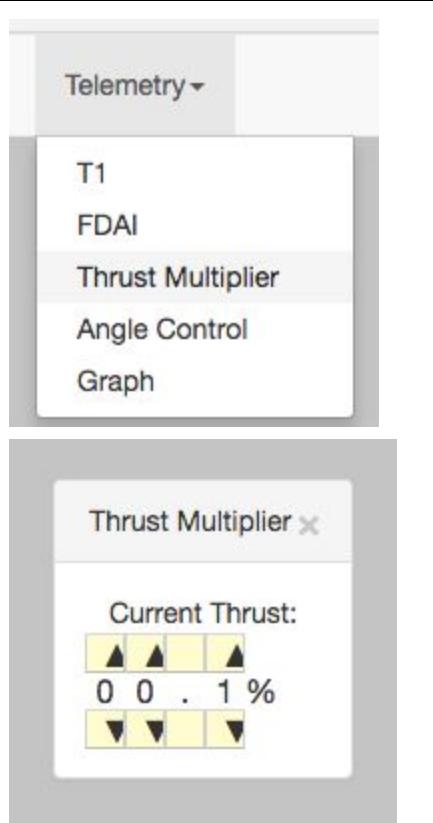
			the server																			
24	Check /lobby renders lobby page correctly	PASS	The lobby.html page should be rendered at /lobby to make sure page routing works for the lobby																			
25	Check lobby displays game settings	PASS	The lobby should show the current game settings to help the players understand how their specific game will work so they can develop a strategy going in	<table border="1"> <tr> <td>Game Name</td> <td>null</td> </tr> <tr> <td>Description</td> <td>null</td> </tr> <tr> <td>Difficulty</td> <td>0</td> </tr> <tr> <td>Online mode</td> <td>true</td> </tr> <tr> <td>White list</td> <td>false</td> </tr> <tr> <td>Cheats</td> <td>false</td> </tr> <tr> <td>Max game length</td> <td>null</td> </tr> <tr> <td>Scinario</td> <td>false</td> </tr> <tr> <td>Active</td> <td>false</td> </tr> </table>	Game Name	null	Description	null	Difficulty	0	Online mode	true	White list	false	Cheats	false	Max game length	null	Scinario	false	Active	false
Game Name	null																					
Description	null																					
Difficulty	0																					
Online mode	true																					
White list	false																					
Cheats	false																					
Max game length	null																					
Scinario	false																					
Active	false																					
26	Check lobby displays currently connected users	PASS	The lobby should show a table of all the currently connected users, to let the players know when to start the game and who they're playing with	<table border="1"> <tr> <td>null</td> <td>Cathryn</td> </tr> <tr> <td>null</td> <td></td> </tr> <tr> <td>0</td> <td></td> </tr> <tr> <td>true</td> <td></td> </tr> <tr> <td>false</td> <td></td> </tr> <tr> <td>false</td> <td></td> </tr> </table>	null	Cathryn	null		0		true		false		false							
null	Cathryn																					
null																						
0																						
true																						
false																						
false																						
27	Check that the server recognises a user has	PASS	Observe server terminal readout and the name of the connected user should be printed when	<pre>(cathy) accepted ('127.0.0.1', 55155) 127.0.0.1 - - [11/Apr/2018 14:46:30] "GET / HTTP/1.1" 2 127.0.0.1 - - [11/Apr/2018 14:46:30] "GET /resources/js Cathryn Cathryn <SecureCookieSession {'username': 'Cathryn'}> 127.0.0.1 - - [11/Apr/2018 14:47:44] "POST /login HTTP/ 127.0.0.1 - - [11/Apr/2018 14:47:44] "GET /lobby HTTP/</pre>																		

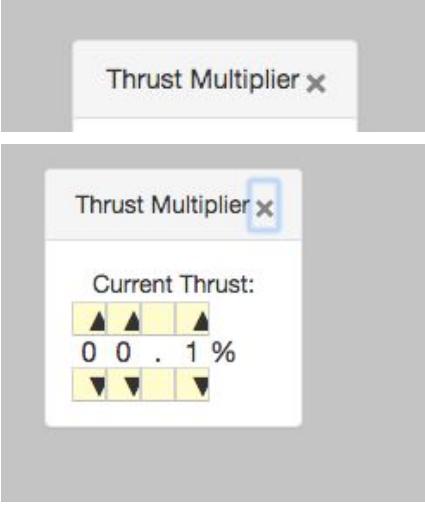
	connected		they connect. This checks that the /lobbu namespace is working and messages can be transmitted on a seperate websocket channel	
28	Check the server allows multiple users to be added	PASS	Each username should be printed in the server terminal in the order they join. This makes sure that multiple users can join	<pre>127.0.0.1 -- [11/Apr/2018 14:48:16] "GET /resource" bob <SecureCookieSession {"username": "bob"}> 127.0.0.1 -- [11/Apr/2018 14:48:24] "POST /login 127.0.0.1 -- [11/Apr/2018 14:46:30] "GET / HTTP/1.1" 2 127.0.0.1 -- [11/Apr/2018 14:46:30] "GET /resources/js Cathryn Cathryn <SecureCookieSession {"username": "Cathryn"}> 127.0.0.1 -- [11/Apr/2018 14:47:44] "POST /login HTTP/ 127.0.0.1 -- [11/Apr/2018 14:47:44] "GET /lobby HTTP/1</pre>
29	All clients update data instantly	PASS	Observe both old and new tabs and both username lists should update simultaneously in real time.	
30	Check client redirects to /ui.html when game starts	PASS	The window should automatically redirect to /ui.html. This test needs to be completed to make sure that the game can be started remotely from the server in the final version.	
31	Check main websocket channel is running correctly	PASS	In the top right the current game time should be continuously running. This means the client and server are communicating over websockets correctly. This check is one of the most important because the entire game relies on the use of this websocket channel so it	

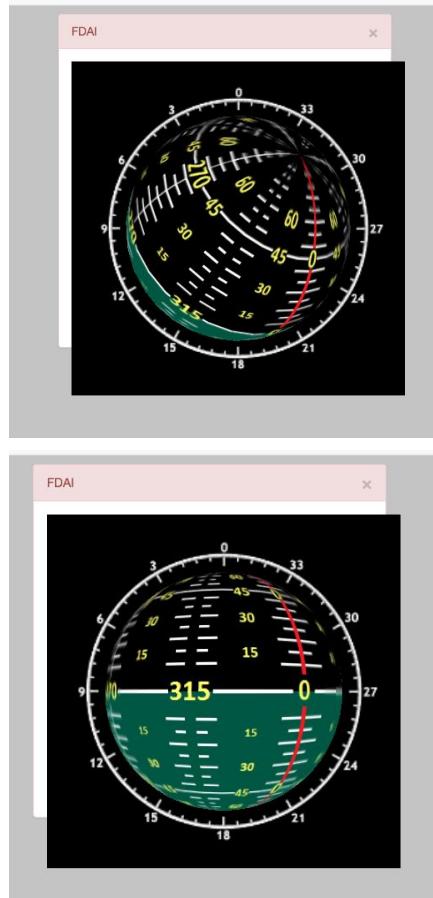
			is very important to check that it works in the final version.	
32	Check /lobbu websocket channel is running correctly	PASS	When the lobby has been joined the username entered in /index.html should appear in the user's list. This string can only be pulled from the server so it indicates that the name has been sent to the server and the server has responded both on the correct /lobbu channel. This check is necessary as it means the namespace routes are set up correctly	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> null null 0 true false false </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Cathryn </div>
33	Check /update websocket channel is running correctly	PASS	When the rocket angle has been changed check the server terminal readout. This should print the data received. The data received should match the new rocket angle and thus proves that the /update channel is working fine. This channel is important because it's the only way the user can interact with the game simulation. Without it there wouldn't be a game.	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> p_pos: be+24,0,19999153932.501 p_sta: true p_thm: 6 p_vel: -22704.150567348806,0,0 </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Thrust Multiplier </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Current Thrust: <div style="display: flex; align-items: center; gap: 10px;"> ▲ ▲ ■ ■ ■ ■ </div> 0 6 . 0 % </div>

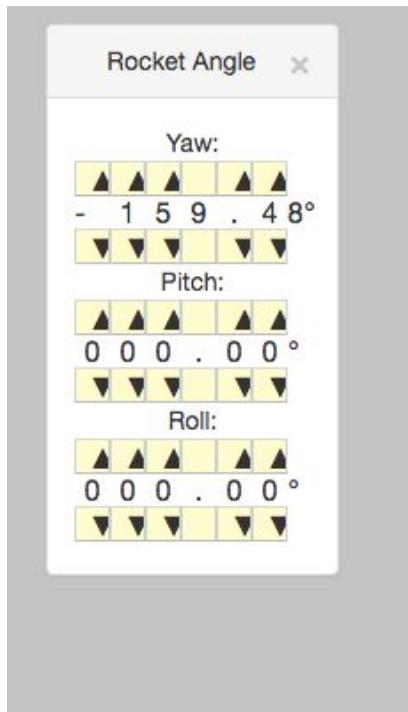
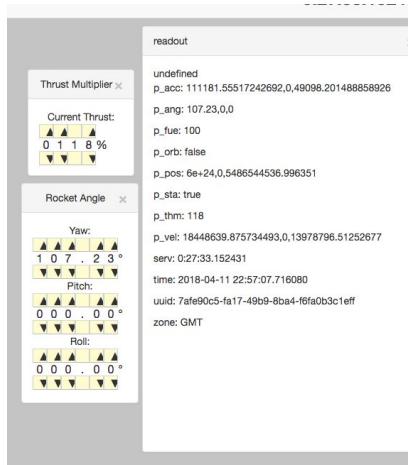
34	In /ui.html, check modals start hidden	PASS	Modals shouldn't be visible to start. This encourages the user to use the navigation options later to set up their own custom workspace	
35	In /ui.html, check modals cannot be dragged outside the visible page boundaries	PASS	The modal edge should bump against the page border edge so no part of the modal is outside the page. This stops a modal being lost or unrecoverable	
36	In /ui.html, check modals become translucent when you dragged	PASS	The modal should show a reduced opacity while dragged to show the user what is underneath whilst rearranging their workspace	
37	In /ui.html, check that modals are draggable	PASS	Modal should move with the cursor. This will improve usability by allowing a custom workspace	
38	In /ui.html, check that modals are only draggable by the top bar	PASS	Modal shouldn't move when dragged by content area. This is important to stop modals being accidentally dragged when interacting with them	
39	In /ui.html, check that modals can overlap	PASS	The modals should have no problem overlapping where one sits on top of another. This improves usability if there is a crowded workspace	

40	In /ui.html, check that the currently dragged modal is in front	PASS	The modal should appear on top of any modals behind so you can see what you're doing	
41	In /ui.html, check that latest dragged modal is on top	PASS	The latest dragged modal should have the highest z-index on the page. This allows the user to define which modals are most important if they've overlapping	
42	In /ui.html, check that multiple modals can be open simultaneously	PASS	More than one modal should be visible at once to allow the user to be able to use multiple modals simultaneously	
43	In /ui.html, check modals can be loaded in any order	PASS	The modal should become visible in the order that the buttons were pressed. This is important for usability	

44	In /ui.html, check that the navigation bar loads	PASS		
45	In /ui.html, check that the navigation bar has a functional drop down menu	PASS	A drop down menu should appear from that specific heading to allow for a more categorical range of options to create a more organised system	
46	In /ui.html, check that the navigation bar buttons can trigger a modal	PASS	The button should trigger the corresponding modal to become visible so it can be used	
48	In /ui.html, check that an exit button appears in the	PASS	An exit [x] button should appear in the top right of the handle. This is an important UI feature to	

	modal handle		make the program easier to use.	
49	In /ui.html, check that the exit button hides the modal	PASS	The Modal should disappear (be hidden) to allow the user to clean up / reorganise their work space	
50	In /ui.html, check that nav-ball can rotate according to defined server data over websockets	PASS	The navball should spin in accordance to websocket data. This ensures that websocket data can be used to influence and change three.js entities.	

				
51	Check numeric variable editor can handle negative values	PASS	The NVE should add a negative sign at the beginning instead of changing one of the digits to a negative sign to avoid confusion for the end user.	

				
52	Check that numerical variable editors for angle changes server data	PASS	Observing the server readout in /ui.html should show that the value (pulled straight from server over websockets) changes in accordance with the NVE value. This is to check that the data can be sent from client and directly edit angle variables in the computational engine (resources.py)	

5.0 Evaluation

5.1 Success Criteria

Success Criteria 1: My system must be networked

Criteria	Element	Has it been met?	Where has it been met?
1	The game must be accessible by a browser	Yes	While the server is running the game (client) is accessible at the server IP where it can be played. Websocket data is sent back and forth from server and client to allow the entire game to take place in a browser window. This is supported by test number 13 and 14 which show clearly that the client (game) is accessible by a browser.
1	The game must be multiplayer	Yes	The game allows for multiple sessions to be connected to the server at once and receive the same data. This allows players to all interact with the same game allowing a multiplayer aspect. The server also keeps track of players names and session IDs to help the multiplayer aspect. This is backed up by tests 28 & 29 which shows that all users see exactly the same information simultaneously and that the server allows and records multiple sessions at once.

1	The system must have a server	Yes	A server is included which serves the game to the user, hosts websocket communications and runs game calculations. This is supported by Test 4 which clearly shows a server starting on the local feedback, confirmed using wireshark.
1	The system must work in real time	Yes	The server runs on multiple threads with the calculations being run continuously with a very accurate time. The client receives new processed data very quickly. The whole system is as real time as possible with this kind of setup. This is supported by test 31 which shows the time running at the same rate as real time.
2	There must be Aerodynamic drag simulations	Yes	Aerodynamic drag is taken into account in the game in resources.py as part of the forces calculation functions. My in development testing supports this.
2	There must be Gravitational simulations	Yes	Newton's laws of gravity are taken into account in the game in resources.py as part of the forces calculation functions. My in development testing supports this
2	There must be orbital mechanical simulations	Yes	Orbital mechanics equations are taken into account in the game in resources.py as part of the orbit check calculation

			to give the game an objective. My in development testing supports this
2	There must be standard rocket calculations	Yes	The “rocket equation” is used to calculate the thrust of the rocket in resources.py as part of the Force calculations. My in-development testing supports this
2	It must feature 3D vector coordinates	Yes	Resources.py stores the player’s position in 3D vector format in an array. My in-dev testing confirms this.
2	It must be able to calculate results iteratively in real time	Yes	The calculations are run in a separate, isolated thread iteratively using a set time base that can be pulled out by the server and sent to the client in real time. Test 31 and 50 shows the client updating in real time with data calculated by the server
3	The game must feature a customisable user interface	Yes	The modals in ui.html are draggable to suit the users preference of arrangement and allows the user to open and close different modals as they require. Tests 34-43 show clearly that modals can be opened and closed as the user wishes and moved around to the most convenient location which supports the fact that it is customisable..
3	The UI must be intuitive and familiar	Yes	The UI features a navbar

			at the top which is a standard in operating systems and websites in 2018 so is familiar. The modals are familiar as they can be compared to windows in any modern Operating system with a GUI. Tests 34-43 show operating system style windows with supports it being familiar and test 45 shows a familiar navbar.
3	The UI must allow for different tools to be used simultaneously	Yes	Multiple modals can be opened at once in ui.html allowing for multiple tools to be accessed at once. Test 42 shows that multiple tools can be used at once.
3	The UI must have an organised way of navigating through different tools	Yes	The UI in ui.html features a navbar with different categories for each tool. Test 45 demonstrates that the navbar categorises tools.
3	The UI must have some visual representations of game data	Partially	ui.html features a navball to indicate direction and has graphing capabilities but the graphing capabilities aren't properly implemented for every variable. A solution where you could select two variables to plot against each other would be a better solution here.
4	The server must feature an admin win “wizard”	Partially	Run.py acts as a server wizard where you can start and stop the server and specify ports and IP address but there isn't much functionality

			beyond this
4	The server must have live feedback of what's happening	Partially	The server gives debugging data in the terminal and tells you when player inputs data, but doesn't give you any live feedback on current variables. However by logging onto the client you can also see the same data that the users see.
5	The game must have an achievable objective	Partially	The game features a orbit_check function which gives the objective to get your rocket into orbit. However at this stage it is the only objective.
5	The game must have an element of user input that directly affects the course of events	Yes	The ui allows users to change angle and thrust modifier variables which are fed back to the server and changed. This is supported by test 52 which shows NVE data changing test data and changing the value of Force and acceleration,
5	The game must have event scenarios	No	The game currently has no capacity to include different scenarios. This could be added very easily
5	The game should include extra non-flight aspects such as life support	No	The game currently includes no extra non-flight aspects however this again could be rectified easily.

Have all success criteria been met?

No. The game i've built doesn't feature an option to include event scenarios, such as faulty equipment, cabin fires, medical issues. This is one of the game mechanics that my end users wanted to be included to make the game more interesting and realistic. Unfortunately due to time constraints this mechanic was never finished, however, it would be very easy to implement.

Scenarios could be triggered in the admin panel using a function that would trigger some variables to change in the resources.py. These variables would act as flags, like 'cabin_fire = true' and to act on these flags the user could change certain environmental variables such as cabin oxygen, and then resources.py would iteratively check the cabin oxygen variable to see if cabin_fire needs to be set false. I would have liked to include a mechanic like this.

This sort of scenario would also fall under the criteria category of non-flight extras, which I also didn't meet. Again this wasn't implemented due to time constraints and would be very easy to implement given more time. Features like life support I started to develop but never had time to actually implement into the game.

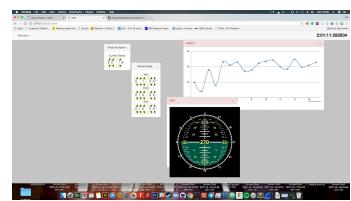
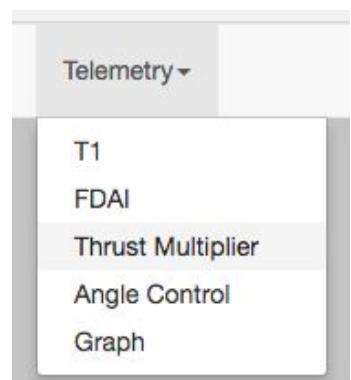
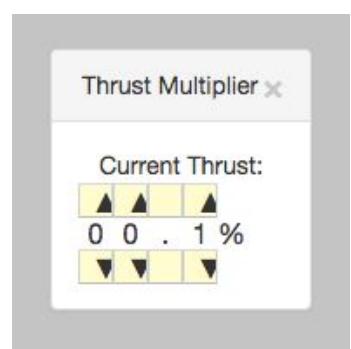
I partially met the criteria for the UI having visual representations of data. In ui.html there are several different graphical tools such as the FDAO (nav-ball) and variable graphs. These tools are incredibly useful and I find paramount to gameplay as I can't understand what's happening without them. It would be nice to add more visual elements such as a rocket visualiser to show the different tanks in the rocket, or an orbit visualiser to show that you are in orbit of a planet. Again this could be added with more time. The server's admin "wizard" has been built but only partially meets the criteria because although you can start and stop the server, start the game and change networking settings, there isn't any functionality implemented to inject information into the game or get information back.

A good idea for this would be a list of users appearing in the admin panel that you could kick from the user rosta if you desired. This hasn't been implemented partially due to time but also because the system I have designed for passing information between the admin panel and the server relies on a text file as an intermediary stage. This is a poor solution but a makeshift one. The idea of this is to get around the subprocess issue of having to Lock a subprocess everytime you want to access data. The problem being that by Locking the server subprocess you stop the server. With more time this would be an issue I would prioritise and try to tackle.

"The server must have live feedback" has been partially met because it does have some feedback such as when users join, when clients load pages, and when users input data and what that data is, but the representation is poor. It can be fully met by adding a second Web UI specifically for the admin rather than a python window. This could communicate over websockets and would likely get around the server lock issue as well. In my final version however if the admin requires to see the game as it's playing they can log on to the client and observe but not interact.

Finally “the game must have an achievable objective” is only partially met because there is only one objective implemented which is to get the rocket into orbit. This can be a bit advanced for beginners but a bit easy for experienced players so it offers no range of difficulty. More objectives could be added in the future in the same way that the orbit_check objective has been implemented, however the admin could choose which objective they want to use before the game starts.

5.2 Usability Features

Feature	Has it been met?	Where has it been met?
Easy to operate, draggable modals(windows) with tools inside.	Yes	
Tools(modals) organised and accessible via Navigation bar	Yes	
Easy numerical value changers	Yes	

5.3 Maintenance issues

As far as I'm aware currently my game has no major maintenance issues. Since the game is almost completely independent of other systems it doesn't rely on much that could be changed. One maintenance issue is that the client loads a lot of resources (such as bootstrap, jquery etc) by Content Delivery Network (CDN) meaning it loads it remotely from another source rather than from the server as packages like three.js are. This could cause issues because the CDN resources are out of my control and could go down, change or be updated at any point which would require the code to be changed to match. To get around this you could download all the CDN resources and save them in the server, and then change the resource addresses in the code.

Another maintenance issue would be keeping up with HTML, CSS, JS and general web browser standards. As the internet progresses and grows languages are changed and browsers update. Some browsers no longer use old CSS or HTML standards. Far into the future the game may not run on modern browsers due to compatibility issues so you would have to download an older browser to run it.

The final maintenance issue is that the server runs in python 3.0. While this shouldn't be a problem as python currently lets you access all the older versions of python, future versions of python such as 3.5 and beyond won't run the server correctly without completely refactoring all the code. To solve this issue you would have to download the older version of python.

Something that would make the code easier to maintain in any of these scenarios is the fact that in my final version all of my code is commented to make it easy for future developers or users to understand. My users have high IT competencies so it is plausible that they would want to read or even edit the code themselves.

To help with this, my variable names are standard format and relate to their purpose well, meaning future developers when maintaining code can find what they need easily.

Finally all my code is written using PEP8 python standards, so the format, indentation, variable capitalisation & names follow an international standard making it easier for other developers to read and interact with.

More information on PEP8 can be found here: <https://www.python.org/dev/peps/pep-0008/>

5.4 Limitations and Improvements

In my Analysis, I wrote a list of limitations. The limitations are still the same as I predicted at the start of my project.

“The User interface will be purely functional and not particularly aesthetically pleasing”

In some areas this isn't true, E.g. the index page where the user signs up looks very nice and had quite a bit of effort put into. However the rest of the system has minimal styling added, for example the lobby is just plaintext and nothing else. The UI has an okay look to it however if I were to have more time I would drastically improve the look of the User Interface to make it more graphical and look more like a space age, sci-fi UX similar to this:

http://test.martiromances.com/?udt_portfolio=the-martian

This limitation was down to time but also it being my lowest priority during development.

Throughout my development I aimed for a functional product. If my stake holders made this a priority it could be fixed however it would take a large amount of time.

“The system will not feature much user management / admin features”

This is still true. There are currently no user management features and admin features are limited. If the stakeholder required this or made it a priority it could be added, however my server would need to focus a lot more on the session IDs of the different clients. I would need to research how this works with Flask and implement it. Then the server would talk to each client session independently. I likely would have to instantiate a new object from a user class everytime a session is created. The admin features would require the admin panel to be web based rather than python based. This would allow the admin panel to interface directly with the server rather than have to deal with complicated sub-process locking protocols.

“The computations will not be perfect and will be heavily abstracted”

This still holds true. I only developed the key elements of the simulation that would make a large effect on the rocket's flight path. Given more time other functions could be implemented into the software easily. The way in which i've built my project allows more functions to be added quickly and still work as the structure already exists to support it. The mathematical functions I included in the simulation also depended on my mathematical skills to derive them. I would likely need extra support from other people when trying to derive complex equations for Aerodynamic heating and surface friction etc.

“The user interface and server will only have a small number of tools and functions”

There are currently only 2 input functions on the user interface and 2 visualising tools so this limitation is still in place. I would ideally like to add more and this would perhaps be what I would consider doing next if I were to continue development. The structure of the system is already

there and ready to support new tools and features, as the server side won't be affected by these new tools at all unless they require new specific data. Most of the visualising requires interpretation of data which is all done client side.

meowterspace / mission_control

Branch: master ▾ mission_control / client / resources / js / navball.js

Find file Copy path

 meowterspace Comments for Final Commit

0f3bcd 39 minutes ago

1 contributor

85 lines (65 sloc) 2.16 KB

```
1 var renderer;
2 var scene;
3 var camera;
4 var cube;
5
6
7 function setup() { // This just sets up the 3D scene and canvas to be the right size.
8
9     // Create an empty scene
10    scene = new THREE.Scene();
11    // Create a basic camera
12    camera = new THREE.PerspectiveCamera(50, 400 / 400, 0.1, 1000);
13    camera.position.z = 10;
14    // Create a renderer with Antialiasing
15    renderer = new THREE.WebGLRenderer();
16    // Configure renderer size
17    renderer.setSize( 400, 400 );
18    // Append Renderer to DOM
19    var canvas = document.getElementById('canvas');
20    canvas.appendChild( renderer.domElement );
21
22 };
23
24
25 function draw(){ // This creates 3D objects within the scene
26
27
28     var geometry = new THREE.SphereGeometry(3, 32, 32); // create a sphere
29     var loader = new THREE.TextureLoader();
30
31     loader.load( // This loads the map.png image to put on the ball.
32         'resources/img/nav-ball/map.png',
33
34         function ( texture ) { // This function sets the texture of the ball
35             // to be the map when the ball has loaded into
36             // the game
37             var material = new THREE.MeshBasicMaterial( {
38                 map: texture
39             } );
40             // defines the ball as 'cube'
41             cube = new THREE.Mesh( geometry, material );
42
43             // Adds the ball with the texture to the scene
44             scene.add( cube );
45
46         },
47         undefined,
48         // error logging
49         function ( err ) {
50             console.error( 'An error happened.' );
51         }
52
53     );
54     // This creates a new 2D plane in the 3D scene
55     var pgeometry = new THREE.PlaneGeometry(9, 9, 32)
56
57     loader.load( // agian, this loads the shader.png texture and applies it to the plane.
58         'resources/img/nav-ball/shader.png',
59         function (texture) {
60             var pmaterial = new THREE.MeshBasicMaterial({
61                 map: texture });
62             plane = new THREE.Mesh( pgeometry, pmaterial );
```

```
63     scene.add( plane ); // add the plane to the 3d scene
64     plane.position.set(0, 0, 0); // set the position of the plane
65     plane.transparent = true;
66   });
67
68 };
69
70 // This renderes the scene
71 var render = function () {
72   requestAnimationFrame( render );
73
74   renderer.render(scene, camera);
75 };
76
77 // This function pulls all the above functions together to render the entire navball at once.
78 // it is called when the FDAI/Navball modal/window is opened in the UI
79 function render_navball() {
80   setup();
81   draw();
82   render();
83 }
84
```

meowterspace / mission_control

Branch: master ▾ [mission_control](#) / [client](#) / [resources](#) / [js](#) / [graphing.js](#)

[Find file](#) [Copy path](#)

 **meowterspace** Comments for Final Commit

0f3bcd 39 minutes ago

1 contributor

34 lines (24 sloc) 782 Bytes

```
1 // This is the graphing script. It currently isn't correctly implemented.
2
3
4 window.onload = function () { // when the window is loaded
5     var chart = new CanvasJS.Chart("chartContainer", { // creat a new canvas
6
7         data: [ // graph starting data. needs to be empty
8             {
9                 type: "spline",
10                dataPoints: [
11                    { y: 10 },
12                    { y: 4 },
13                    { y: 18 },
14                    { y: 8 }
15                ]
16            }
17        ]
18    });
19    chart.render(); // render the graph in the scene
20
21
22 // This function will be connected to the websocket recieve message function.
23 // It adds a new point to the graph defined by {}. Very easy to implement.
24
25 $("#addDataPoint").click(function () {
26
27     var length = chart.options.data[0].dataPoints.length;
28     chart.options.data[0].dataPoints.push({ y: 25 - Math.random() * 10});
29     chart.render();
30
31 });
32
33
34 }
```

meowterspace / mission_control

Branch: master [mission_control](#) / client / resources / js / anykey.js

[Find file](#) [Copy path](#)

 **meowterspace** Comments for Final Commit

0f3bcd 39 minutes ago

1 contributor

Executable File 28 lines (26 sloc) 3.16 KB

```
1 var menu_flag = 0; // stores if the Press any key to continue function has already been activated
2
3 $(document).on('keyup',function(evt) { // if a keyup event occurs
4     if (menu_flag == 0){ // and if it hasn't occurred before
5         if (evt.keyCode == 8||evt.keyCode == 9||evt.keyCode == 13||evt.keyCode == 16||evt.keyCode == 17||evt.keyCode == 18||evt
6             // ^ if it is a standard ascii keyboard key,
7             $('#menu').show(); // show the menu
8             $('#PAKTC').hide(); // and hide the press any key to continue text
9             menu_flag = 1; // set the menu flag so it can't happen again.
10        };
11    };
12 });
13
14 // This function is called by the join button. It hides the menu and shows the join menu.
15 function join() {
16     $('#menu').hide();
17     $('#join-menu').show();
18 };
19 // this function is called by the host button. It displays some text then redirects the user to the github
20 // repo so they can look into hosting their own game.
21 function host() {
22     alert('You will now be redirected to the GitHub repo for MCS 3000'); // displays redirect alert
23     window.location.replace('http://github.com/meowterspace/mission_control') // redirects to repo
24 };
25 // This function is called by the about button. It displays some very basic about text.
26 function about() {
27     alert('This is the mission control simulator 3000. A multiplayer simulation that allows you to take a seat in mission c
28 };
```

meowterspace / mission_control

Branch: master ▾ [mission_control / client / index.html](#)

[Find file](#) [Copy path](#)

 **meowterspace** Comments for Final Commit

0f3bcd 38 minutes ago

1 contributor

Executable File 158 lines (136 sloc) 42.5 KB

```
1  <!DOCTYPE html>
2  <html >
3  <head>
4      <meta charset="UTF-8">
5      <title>Parallax Star background in CSS</title>
6
7      <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/1.4.8/socket.io.min.js"></script>
8
9      <script src="https://s.codepen.io/assets/libs/modernizr.js" type="text/javascript"></script>
10     <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
11     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css" integrity="sha384-
12     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css">
13
14
15     <style>
16         html {
17             height: 100%;
18             background: radial-gradient(ellipse at bottom, #1b2735 0%, #090a0f 100%);
19             overflow: hidden;
20         }
21
22 #stars {
23     width: 1px;
24     height: 1px;
25     background: transparent;
26     box-shadow: 903px 1229px #FFF , 507px 1263px #FFF , 1080px 538px #FFF , 155px 1608px #FFF , 112px 1567px #FFF , 195px 1511px
27     animation: animStar 50s linear infinite;
28 }
29 #stars:after {
30     content: " ";
31     position: absolute;
32     top: 2000px;
33     width: 1px;
34     height: 1px;
35     background: transparent;
36     box-shadow: 903px 1229px #FFF , 507px 1263px #FFF , 1080px 538px #FFF , 155px 1608px #FFF , 112px 1567px #FFF , 195px 1511px
37 }
38
39 #stars2 {
40     width: 2px;
41     height: 2px;
42     background: transparent;
43     box-shadow: 1583px 294px #FFF , 547px 1420px #FFF , 88px 1212px #FFF , 544px 691px #FFF , 578px 469px #FFF , 735px 1350px #FF
44     animation: animStar 100s linear infinite;
45 }
46 #stars2:after {
47     content: " ";
48     position: absolute;
49     top: 2000px;
50     width: 2px;
51     height: 2px;
52     background: transparent;
53     box-shadow: 1583px 294px #FFF , 547px 1420px #FFF , 88px 1212px #FFF , 544px 691px #FFF , 578px 469px #FFF , 735px 1350px #FF
54 }
55
56 #stars3 {
57     width: 3px;
58     height: 3px;
59     background: transparent;
60     box-shadow: 396px 818px #FFF , 1349px 785px #FFF , 727px 524px #FFF , 1952px 1580px #FFF , 1469px 342px #FFF , 1571px 1615px
61     animation: animStar 150s linear infinite;
62 }
```

```
63 #stars3:after {
64   content: " ";
65   position: absolute;
66   top: 2000px;
67   width: 3px;
68   height: 3px;
69   background: transparent;
70   box-shadow: 396px 818px #FFF , 1349px 785px #FFF , 727px 524px #FFF , 1952px 1580px #FFF , 1469px 342px #FFF , 1571px 1615px
71 }
72
73 #title {
74   position: absolute;
75   top: 50%;
76   left: 0;
77   right: 0;
78   color: #FFF;
79   text-align: center;
80   font-family: "lato", sans-serif;
81   font-weight: 300;
82   font-size: 50px;
83   letter-spacing: 10px;
84   margin-top: -60px;
85   padding-left: 10px;
86 }
87 #title span {
88   background: -webkit-linear-gradient(white, #38495a);
89   -webkit-background-clip: text;
90   -webkit-text-fill-color: transparent;
91 }
92
93 @keyframes animStar {
94   from {
95     transform: translateY(0px);
96   }
97   to {
98     transform: translateY(-2000px);
99   }
100 }
101 /* MODAL STUFF */
102
103 .menu {
104   position: absolute;
105   z-index: 1000;
106   display: none;
107
108   top: 65%;
109 }
110
111 </style>
112
113 <script src="https://cdnjs.cloudflare.com/ajax/libs/prefixfree/1.0.7/prefixfree.min.js"></script>
114
115 </head>
116
117 <body>
118
119
120   <script src="resources/js/anykey.js"></script>
121   <link href='https://fonts.googleapis.com/css?family=Lato:300,400,700' rel='stylesheet' type='text/css'>
122 <div id='stars'></div>
123 <div id='stars2'></div>
124 <div id='stars3'></div>
125 <div id='title'>
126   <span id="text">
127     MISSION CONTROL SIM
128   </span>
129   <br>
130   <span id="text">
131     3000
132   </span>
133   <br>
134   <br>
135   <div id='PAKTC'><span id="text" style="font-size:30px;">PRESS ANY KEY TO START</span></div>
136 </div>
```

```
137 </div>
138 <script src='http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></script>
139
140 <div class="menu" id="menu" style="width:100%"><h1 style='color:white;'><center>
141   <button onclick="join()" type="button" class="btn btn-secondary" style="width:75px">Join</button><br>
142   <button onclick="host()" type="button" class="btn btn-secondary" style="width:75px">Host</button><br>
143   <button onclick="about()" type="button" class="btn btn-secondary" style="width:75px">About</button></center>
144 </div>
145 <div class="menu" id="join-menu" style="width:100%"><h1 style='color:white;'>
146   <center>
147     <form action="/login" method="POST">
148       <input type="text" name="username" placeholder="Display Name" style="text-align: center;"><br><br>
149       <button type="submit" class="btn btn-secondary" style="width:75px;">Join</button><br>
150     </form>
151   </center>
152 </div>
153
154
155
156 </body>
157 </html>
```

meowterspace / mission_control

Branch: master ▾ mission_control / client / lobby.html

Find file Copy path

 meowterspace Comments for Final Commit

0f3bcd 38 minutes ago

1 contributor

55 lines (46 sloc) 2.24 KB

```
1 <head>
2 <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
3 <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
4 <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/1.4.8/socket.io.min.js"></script>
5 </head>
6 <body>
7 <script type="text/javascript">
8 $(document).ready(function() {
9     var lobby = io.connect('http://127.0.0.1:5000/lobbu'); // connect to websocket channel /lobbu
10
11
12     lobby.on('connect', function() { // when the page connects to the websocket server:
13         lobby.send('hi'); // send a message 'hi'. This will start the back and forth communication
14
15
16 });
17
18     lobby.on('message', function(msg) { // When this receives a message from /lobbu channel,
19         if (String(msg.ACTIVE) == 'true') { // if the game is active, redirect to /ui.html
20             window.location.replace("http://127.0.0.1:5000/ui.html");
21         }
22
23         // The below prints out the websocket data in two tables.
24         // This table is the people in the game
25         var user_table = '<table style="border: 1px solid #000000">';
26         for (var i=1; i <= msg.USER_LIST.length; i++) {
27             user_table = user_table + '<tr><td>' + msg.USER_LIST[i-1] + '</td></tr>';
28         }
29         user_table = user_table + '</table>';
30         // Add table to document
31         document.getElementById('party').innerHTML = user_table;
32         lobby.send('hi');
33         // This table is the current game settings
34         var settings_table = '<table style="border: 1px solid #000000">
35             <tr><td>Game Name</td><td>' + String(msg.GAME_NAME) + '</td></tr>
36             <tr><td>Description</td><td>' + String(msg.DESCRIPTION) + '</td></tr>
37             <tr><td>Difficulty</td><td>' + String(msg.DIFFICULTY) + '</td></tr>
38             <tr><td>Online mode</td><td>' + String(msg.ONLINE_MODE) + '</td></tr>
39             <tr><td>White list</td><td>' + String(msg.WHITE_LIST) + '</td></tr>
40             <tr><td>Cheats</td><td>' + String(msg.CHEATS) + '</td></tr>
41             <tr><td>Max game length</td><td>' + String(msg.MAX_GAME_LENGTH) + '</td></tr>
42             <tr><td>Scinario</td><td>' + String(msg.SCINARIO) + '</td></tr>
43             <tr><td>Active</td><td>' + String(msg.ACTIVE) + '</td></tr>
44
45         ';
46
47         // Add table to document
48         document.getElementById('settings').innerHTML = settings_table;
49     });
50 });
51 </script>
52 <div style="float:left" id='settings'></div>
53 <div style="float:left; padding-left:10px" id='party'></div>
54
55 </body>
```

 meowterspace Comments for Final Commit

0f3bcd 38 minutes ago

1 contributor

421 lines (370 sloc) 15 KB

```
1 <html>
2     <head>
3
4         <title>Main</title>
5         <!-- INCLUDE THE SOCKETIO LIBS VIA CDN -->
6         <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/1.4.8/socket.io.min.js"></script>
7         <!-- INCLUDE THE JQUERY LIBRARIES VIA CDN -->
8         <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
9         <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
10
11         <!-- INCLUDE THE BOOTSTRAP LIBRARIES VIA CDN -->
12         <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
13         <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
14         <script type="text/javascript" src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>
15         <script src="resources/js/three.min.js"></script>
16         <script src='resources/js/navball.js'></script>
17         <script src='resources/js/graphing.js'></script>
18         <style type="text/css">
19             body {
20                 background-color: #c4c4c4;
21                 margin: 0px;
22                 height: 100%;
23             }
24             #win1 {
25                 width: 400px;
26                 height: 600px;
27             }
28             #win-thm {
29                 width: 150px;
30                 height: 150px;
31                 text-align: center;
32             }
33             #win-ang {
34                 width: 165px;
35                 height: 310px;
36                 text-align: center;
37             }
38             #win-FDAI {
39                 width: 400px;
40                 height: 400px;
41                 margin: 0;
42             }
43             #win-graph {
44                 width: 800px;
45                 height: 400px;
46                 margin: 0;
47             }
48             .panel {
49                 position: absolute;
50                 clear: both;
51                 display: none;
52             }
53             #canvas {
54                 width: 100%;
55                 height: 100%;
56
57             }
58             td{
59                 text-align: center;
60             }
61
62             input{
```

```

63             text-align: left;
64         }
65
66         #up{
67             width: 20px;
68             height: 20px;
69             background-color: #ffffdd0;
70         }
71
72
73     </style>
74
75     <script>
76         var FDAI_rendered = 0; // marker to show if the FDAI has been rendered
77     </script>
78
79     <script>
80
81         var latest_data = null; // Latest data received by websockets
82
83
84         // ===== EDIT STUFF =====
85         var update = io.connect('http://127.0.0.1:5000/update'); // connect to /update websocket channel
86
87         /* The function below is used to display a value on a Numerical Value Editor.
88             it takes the id's it needs to edit in the document, the value it wants to display
89             and the position in the value of the decimal point. */
90         function make(id, value, dp){ // List of IDs for values, value, decimal point position in array
91             value = Math.round(value*100)/100; // round the value to 2dp
92             console.log(value);
93             value = value.toString();
94             value = value.split(''); // split each character into a new element in the array
95             if (value.length < id.length ) {value.unshift('0');};
96             if (value[dp] == null){ value[dp] = '.'; } // This section just makes sure nothing is
97             for (var i=0; i<id.length; i++){ // null and will fix it if it is.
98                 if (value[i] == null){ value[i] = '0'; };
99
100             document.getElementById(id[i]).innerHTML = value[i]; // add each value to each ID in document
101         };
102     };
103
104
105     /* This function is used to change the value on a Numerical Value Editor. It takes the type of NVE,
106         the power of the unit it should change and the direction (positive or negative). It then calculates
107         the new value, updates latest_data and calls the make function above to display the new value */
108     function change(type, power, direction){
109         console.log(value, power, direction);
110         switch(type){
111             case 'thm': // The cases represent different NVEs on the screen. This is for the Thrust Modifie
112                 var value = latest_data.p_thm;
113                 value = value + (1*power*direction);
114                 make(['thm1', 'thm2', 'thm3', 'thm4'], value, 2);
115                 latest_data.p_thm = Math.round(value*100)/100;
116                 break;
117             case 'angy': // The yaw angle
118                 var value = latest_data.p_ang[0];
119                 value = value + (1*power*direction); // change value to match new data that user select
120                 make(['angy1', 'angy2', 'angy3', 'angy4', 'angy5', 'angy6'], value, 3); // update displ
121                 latest_data.p_ang[0] = Math.round(value*100)/100; // update latest_data
122                 break;
123             case 'angp': // The Pitch angle
124                 var value = latest_data.p_ang[1];
125                 value = value + (1*power*direction);
126                 make(['angp1', 'angp2', 'angp3', 'angp4', 'angp5', 'angp6'], value, 3);
127                 latest_data.p_ang[1] = Math.round(value*100)/100;
128                 break;
129             case 'angr': // The Roll angle
130                 var value = latest_data.p_ang[2];
131                 value = value + (1*power*direction);
132                 make(['angr1', 'angr2', 'angr3', 'angr4', 'angr5', 'angr6'], value, 3);
133                 latest_data.p_ang[2] = Math.round(value*100)/100;
134                 break;
135         };
136         // UPDATE PAGE DATA HERE

```

```

137
138         update.send(latest_data); // send the newly updated data(latest_data) to the server on the /update name
139     };
140
141
142
143     /* This makes #winx dragabble only by handle(class .handle)
144     with a change in opacity on event and ensures most recently dragged
145     has the highest z-index */
146     var zIndex = 1;
147     $(function() {
148
149         $("div[id*='win']").draggable({
150             containment: "window",
151             scroll: false,
152             handle: $(".handle"),
153             opacity: 0.8,
154             start: function(event, ui) {
155                 $(this).css("z-index", ++zIndex);
156
157             }
158         });
159     });
160
161     /* Function for changing css attribute display to true.
162     On show increases z-index to be the highest of all windows
163     to prevent displacement */
164     function showModal(id) {
165         var win = document.getElementById(id);
166         $(win).css("z-index", ++zIndex);
167         $(win).show();
168         if (id=='win-fdai' && FDAI_rendered >= 0) {
169             render_navball();
170             FDAI_rendered = 1;
171         }
172     };
173
174     /* Function for hiding the modal */
175     function hideModal(id) {
176         var win = document.getElementById(id);
177         $(win).hide();
178     };
179
180
181     </script>
182
183     </head>
184
185     <body>
186
187     <script type="text/javascript">
188     $(document).ready(function() {
189         var socket = io.connect('http://127.0.0.1:5000'); // connect to main websocket channel
190
191
192         socket.on('connect', function() { // when it connects to server, send a message to get the data flow going
193             socket.send('User has connected!');
194         });
195
196         socket.on('message', function(msg) { // when client receives a message
197
198             latest_data = msg; // store the message as latest_data
199
200             var readout; // This prints out all the data in the message to T1 in the navbar for debug purposes
201                 // or just to look at all the variables at once to get a more comprehensive view of the
202                 // simulation.
203             document.getElementById('serv').innerHTML = msg.serv;
204             readout = readout + '<p>p_acc: '+String(msg.p_acc)+'</p>';
205             readout = readout + '<p>p_ang: '+String(msg.p_ang)+'</p>';
206             readout = readout + '<p>p_fue: '+String(msg.p_fue)+'</p>';
207             readout = readout + '<p>p_orb: '+String(msg.p_orb)+'</p>';
208             readout = readout + '<p>p_pos: '+String(msg.p_pos)+'</p>';
209             readout = readout + '<p>p_sta: '+String(msg.p_sta)+'</p>';
210             readout = readout + '<p>p_thm: '+String(msg.p_thm)+'</p>';

```

```

211     readout = readout + '<p>p_vel: '+String(msg.p_vel)+'</p>';
212     readout = readout + '<p>serv: '+String(msg.serv)+'</p>';
213     readout = readout + '<p>time: '+String(msg.time)+'</p>';
214     readout = readout + '<p>uuid: '+String(msg.uuid)+'</p>';
215     readout = readout + '<p>zone: '+String(msg.zone)+'</p>';
216     document.getElementById('readout').innerHTML = readout;
217
218     // This checks if the FDAI (navball) has been rendered. If it has it will update it's angle/rotation
219     // to match the simulations angle.
220     document.getElementById('serv').innerHTML = msg.serv;
221     if (FDI_rendered == 1) {
222         FDI_rendered = FDI_rendered + 1;
223     } else if (FDI_rendered == 2) {
224         cube.rotation.x = msg.angle[0];
225     };
226     socket.send(msg.uuid);
227
228 });
229
230 </script>
231
232
233
234     <nav class="navbar navbar-default">
235         <div class="container-fluid">
236             <ul class="nav navbar-nav">
237                 <li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown" href="#">Telemetry</a>
238                     <ul class="dropdown-menu">
239                         <li><a href="javascript:showModal('win1')">T1</a></li>
240                         <li><a href="javascript:showModal('win-fdai')">FDI</a></li>
241                         <li><a href="javascript:showModal('win-thm')">Thrust Multiplier</a></li>
242                         <li><a href="javascript:showModal('win-ang')">Angle Control</a></li>
243                         <li><a href="javascript:showModal('win-graph')">Graph</a></li>
244                     </ul>
245                 </li>
246             </ul>
247             <div style="top:5px; float:right; position: relative;">
248                 <span id='serv' style="font-weight:bold; font-size:25px;">OFFLINE</span>
249             </div>
250         </div>
251     </nav>
252     <div id="win1" class="col-3 panel panel-default">
253         <div class="handle panel-heading">readout
254             <button type="button" class="close" aria-label="Close" onclick="hideModal('win1')">
255                 <span aria-hidden="true">&times;</span>
256             </button>
257         </div>
258         <div class="panel-body">
259             <span id="readout">OFFLINE</span>
260         </div>
261     </div>
262
263     <div id="win-fdai" class="col-2 panel panel-danger">
264         <div class="handle panel-heading">FDI
265             <button type="button" class="close" aria-label="Close" onclick="hideModal('win-fdai')">
266                 <span aria-hidden="true">&times;</span>
267             </button>
268         </div>
269         <div class="panel-body"><div id='canvas'></div></div>
270     </div>
271
272     <!-- ===== THM ===== -->
273
274     <div id="win-thm" class="col-3 panel panel-default">
275         <div class="handle panel-heading">Thrust Multiplier
276             <button type="button" class="close" aria-label="Close" onclick="hideModal('win-thm')">
277                 <span aria-hidden="true">&times;</span>
278             </button>
279         </div>
280         <div class="panel-body">
281             Current Thrust:
282             <table>
283                 <tr>
284                     <td><input type='button' onclick="change('thm', 10, 1)" id='up' value='▲'></td>

```

```

285             <td><input type='button' onclick="change('thm', 1, 1)" id='up' value='▲'></td>
286             <td><input type='button' id='up'></td>
287             <td><input type='button' onclick="change('thm', 0.1, 1)" id='up' value='▲'></td>
288         </tr>
289         <tr id='text'>
290             <td id='thm1'>0</td>
291             <td id='thm2'>0</td>
292             <td id='thm3'>.</td>
293             <td id='thm4'>0</td>
294             <td>%</td>
295         </tr>
296         <tr>
297             <td><input type='button' onclick="change('thm', 10, -1)" id='up' value='▼'></td>
298             <td><input type='button' onclick="change('thm', 1, -1)" id='up' value='▼'></td>
299             <td><input type='button' id='up'></td>
300             <td><input type='button' onclick="change('thm', 0.1, -1)" id='up' value='▼'></td>
301         </tr>
302         </table>
303     </div>
304 </div>
305
306 <!-- ===== ANGLE ===== -->
307 <div id="win-ang" class="col-3 panel panel-default">
308     <div class="handle panel-heading">Rocket Angle
309         <button type="button" class="close" aria-label="Close" onclick="hideModal('win-ang')">
310             <span aria-hidden="true">&times;</span>
311         </button>
312     </div>
313     <div class="panel-body">
314         Yaw:
315         <table>
316             <tr>
317                 <td><input type='button' onclick="change('angy', 100, 1)" id='up' value='▲'></td>
318                 <td><input type='button' onclick="change('angy', 10, 1)" id='up' value='▲'></td>
319                 <td><input type='button' onclick="change('angy', 1, 1)" id='up' value='▲'></td>
320                 <td><input type='button' id='up'></td>
321                 <td><input type='button' onclick="change('angy', 0.1, 1)" id='up' value='▲'></td>
322                 <td><input type='button' onclick="change('angy', 0.01, 1)" id='up' value='▲'></td>
323             </tr>
324             <tr id='text'>
325                 <td id='angy1'>0</td>
326                 <td id='angy2'>0</td>
327                 <td id='angy3'>0</td>
328                 <td id='angy4'>.</td>
329                 <td id='angy5'>0</td>
330                 <td id='angy6'>0</td>
331                 <td>°</td>
332             </tr>
333             <tr>
334                 <td><input type='button' onclick="change('angy', 100, -1)" id='up' value='▼'></td>
335                 <td><input type='button' onclick="change('angy', 10, -1)" id='up' value='▼'></td>
336                 <td><input type='button' onclick="change('angy', 1, -1)" id='up' value='▼'></td>
337                 <td><input type='button' id='up'></td>
338                 <td><input type='button' onclick="change('angy', 0.1, -1)" id='up' value='▼'></td>
339                 <td><input type='button' onclick="change('angy', 0.01, -1)" id='up' value='▼'></td>
340             </tr>
341             </table>
342         Pitch:
343         <table>
344             <tr>
345                 <td><input type='button' onclick="change('angp', 100, 1)" id='up' value='▲'></td>
346                 <td><input type='button' onclick="change('angp', 10, 1)" id='up' value='▲'></td>
347                 <td><input type='button' onclick="change('angp', 1, 1)" id='up' value='▲'></td>
348                 <td><input type='button' id='up'></td>
349                 <td><input type='button' onclick="change('angp', 0.1, 1)" id='up' value='▲'></td>
350                 <td><input type='button' onclick="change('angp', 0.01, 1)" id='up' value='▲'></td>
351             </tr>
352             <tr id='text'>
353                 <td id='angp1'>0</td>
354                 <td id='angp2'>0</td>
355                 <td id='angp3'>0</td>
356                 <td id='angp4'>.</td>
357                 <td id='angp5'>0</td>
358                 <td id='angp6'>0</td>

```

```

359             <td>°</td>
360         </tr>
361         <tr>
362             <td><input type='button' onclick="change('angp', 100, -1)" id='up' value='▼'></td>
363             <td><input type='button' onclick="change('angp', 10, -1)" id='up' value='▼'></td>
364             <td><input type='button' onclick="change('angp', 1, -1)" id='up' value='▼'></td>
365             <td><input type='button' id='up'></td>
366             <td><input type='button' onclick="change('angp', 0.1, -1)" id='up' value='▼'></td>
367             <td><input type='button' onclick="change('angp', 0.01, -1)" id='up' value='▼'></td>
368         </tr>
369     </table>
370     Roll:
371     <table>
372         <tr>
373             <td><input type='button' onclick="change('angr', 100, 1)" id='up' value='▲'></td>
374             <td><input type='button' onclick="change('angr', 10, 1)" id='up' value='▲'></td>
375             <td><input type='button' onclick="change('angr', 1, 1)" id='up' value='▲'></td>
376             <td><input type='button' id='up'></td>
377             <td><input type='button' onclick="change('angr', 0.1, 1)" id='up' value='▲'></td>
378             <td><input type='button' onclick="change('angr', 0.01, 1)" id='up' value='▲'></td>
379         </tr>
380         <tr id='text'>
381             <td id='angr1'>0</td>
382             <td id='angr2'>0</td>
383             <td id='angr3'>0</td>
384             <td id='angr4'>.</td>
385             <td id='angr5'>0</td>
386             <td id='angr6'>0</td>
387             <td>°</td>
388         </tr>
389         <tr>
390             <td><input type='button' onclick="change('angr', 100, -1)" id='up' value='▼'></td>
391             <td><input type='button' onclick="change('angr', 10, -1)" id='up' value='▼'></td>
392             <td><input type='button' onclick="change('angr', 1, -1)" id='up' value='▼'></td>
393             <td><input type='button' id='up'></td>
394             <td><input type='button' onclick="change('angr', 0.1, -1)" id='up' value='▼'></td>
395             <td><input type='button' onclick="change('angr', 0.01, -1)" id='up' value='▼'></td>
396         </tr>
397     </table>
398 </div>
399 </div>
400 </body>
401
402 <!-- ===== Graph ===== -->
403
404     <div id="win-graph" class="col-2 panel panel-danger">
405         <div class="handle panel-heading">GRAPH
406             <button type="button" class="close" aria-label="Close" onclick="hideModal('win-graph')">
407                 <span aria-hidden="true">&times;</span>
408             </button>
409         </div>
410         <div class="panel-body">
411             <div id="chartContainer" style="width:100%; height:300px"></div>
412             <button id="addDataPoint">Add Data Point</button>
413             <button id="updateDataPoint">Update Data Point</button>
414
415         </div>
416     </div>
417
418
419
420

```

meowterspace / mission_control

Branch: master ▾ [mission_control](#) / server / CONFIG.txt

[Find file](#) [Copy path](#)

 **meowterspace** Added Lobby

c0a71ee on 28 Jan

1 contributor

29 lines (28 sloc) 786 Bytes

```
1 JSON_AS_ASCII = True
2 USE_X_SENDFILE = False
3 SESSION_COOKIE_PATH = None
4 SESSION_COOKIE_DOMAIN = None
5 SESSION_COOKIE_NAME = session
6 DEBUG = False
7 LOGGER_HANDLER_POLICY = always
8 LOGGER_NAME = None
9 SESSION_COOKIE_SECURE = False
10 SECRET_KEY = None
11 EXPLAIN_TEMPLATE_LOADING = False
12 MAX_CONTENT_LENGTH = None
13 PROPAGATE_EXCEPTIONS = None
14 APPLICATION_ROOT = None
15 SERVER_NAME = None
16 PREFERRED_URL_SCHEME = http
17 JSONIFY_PRETTYPRINT_REGULAR = True
18 TESTING = False
19 PERMANENT_SESSION_LIFETIME = 31 days, 0:00:00
20 TEMPLATES_AUTO_RELOAD = None
21 TRAP_BAD_REQUEST_ERRORS = False
22 JSON_SORT_KEYS = True
23 JSONIFY_MIMETYPE = application/json
24 SESSION_COOKIE_HTTPONLY = True
25 SEND_FILE_MAX_AGE_DEFAULT = 12:00:00
26 PRESERVE_CONTEXT_ON_EXCEPTION = None
27 SESSION_REFRESH_EACH_REQUEST = True
28 TRAP_HTTP_EXCEPTIONS = False
```

meowterspace / mission_control

Branch: master ▾ [mission_control](#) / [server](#) / [app.py](#)

[Find file](#) [Copy path](#)

 [meowterspace](#) Comments for Final Commit

0f3bcd 37 minutes ago

1 contributor

241 lines (201 sloc) 9.6 KB

```
1 # Import all external libraries
2 import json
3 import datetime
4 import time
5 import os
6 import io
7 import errno
8 from flask import Flask, render_template, send_file, request, session, redirect, render_template_string
9 from flask_socketio import SocketIO, emit, send
10 from threading import Thread, Lock
11 import resources
12 import sys
13
14 print(sys.argv)
15
16
17 # This is a dictionary of all the settings Flask has. Since I have a complex setup it is best to have the
18 # config ready at all times so I can quickly change server settings. This dictionary is a default dictionary
19 # and will be overwritten when the server starts with the settings specified in CONFIG.txt
20 CONFIG = {
21     'JSON_AS_ASCII': True,
22     'USE_X_SENDFILE': False,
23     'SESSION_COOKIE_PATH': None,
24     'SESSION_COOKIE_DOMAIN': None,
25     'SESSION_COOKIE_NAME': 'session',
26     'DEBUG': False,
27     'LOGGER_HANDLER_POLICY': 'always',
28     'LOGGER_NAME': None,
29     'SESSION_COOKIE_SECURE': False,
30     'SECRET_KEY': None,
31     'EXPLAIN_TEMPLATE_LOADING': False,
32     'MAX_CONTENT_LENGTH': None,
33     'PROPAGATE_EXCEPTIONS': None,
34     'APPLICATION_ROOT': None,
35     'SERVER_NAME': None,
36     'PREFERRED_URL_SCHEME': 'http',
37     'JSONIFY_PRETTYPRINT_REGULAR': True,
38     'TESTING': False,
39     'PERMANENT_SESSION_LIFETIME': datetime.timedelta(31),
40     'TEMPLATES_AUTO_RELOAD': None,
41     'TRAP_BAD_REQUEST_ERRORS': False,
42     'JSON_SORT_KEYS': True,
43     'JSONIFY_MIMETYPE': 'application/json',
44     'SESSION_COOKIE_HTTPONLY': True,
45     'SEND_FILE_MAX_AGE_DEFAULT': datetime.timedelta(0, 43200),
46     'PRESERVE_CONTEXT_ON_EXCEPTION': None,
47     'SESSION_REFRESH_EACH_REQUEST': True,
48     'TRAP_HTTP_EXCEPTIONS': False
49 }
50
51 # This library contains game data, such as the name, and whether or not the game is active. A lot of this isn't
52 # currently implemented however it outlines a plan of future development. Some elements like USER_LIST and
53 # ACTIVE are used throughout the program.
54 GAME = {
55     'GAME_NAME': None,
56     'DESCRIPTION': None,
57     'DIFFICULTY': 0,
58     'ONLINE_MODE': True,
59     'WHITE_LIST': False,
60     'BLACK_LIST': False,
61     'CHEATS': False,
62     'MAX_GAME_LENGTH': None,
```

```

63     'BANNED_USERNAMES': False,
64     'SCENARIO': False,
65     'ACTIVE': False,
66     'USER_LIST': []
67 }
68
69 users = [] # A list of all the current users in the game.
70
71 # This defines the user class.
72 class User:
73     def __init__(self, name):
74         self.name = name
75
76
77 #The code below is responsible for loading in the server settings from CONFIG.txt
78 if os.path.exists('CONFIG.txt') == True: # checks if CONFIG.txt exists yet
79     with open('CONFIG.txt', 'r') as f: # if it does exist, open the file
80         for line in f.readlines():
81             line_split = line.strip().split("=") # For every line, split the left and right half at the = sign
82             CONFIG[line_split[0].strip()] = line_split[1].strip() # Update CONFIG dictionary
83     f.close() # close CONFIG.txt
84 else: # if it doesn't exist
85     with open('CONFIG.txt', 'w') as f: # Open CONFIG.txt (generates file)
86         for key in CONFIG:
87             f.write(key+' = '+str(CONFIG[key])+'\n') # write the default config to the file
88     f.close() # Close new file
89
90
91 # Flask server setup
92 app = Flask(__name__, template_folder='../client') # Defines the Flask Server. The template_folder is set to
93 # ../client (the client folder) which means the browser has
94 # no access to the server at all as it is in a different
95 # directory.
96 app.config['SECRET_KEY'] = 'secret' # Sets the server encryption key
97
98 app.config['DEBUG'] = CONFIG['DEBUG'] # Sets the server settings to equal CONFIG
99
100
101 for i in app.config: print(i, app.config[i]) # prints the new server CONFIG to the terminal
102
103 socketio = SocketIO(app) # Defines SOCKETIO app
104
105 # Below is a meta data dictionary that is to be updated and sent with each websocket message
106 # to the client so the message can be verified for debugging / tracking purposes, or
107 # can be used to measure information loss in the websocket stream
108 meta = {
109     'time' : '', #current Time
110     'zone' : 'GMT', #timezone
111     'serv' : '', #time the server's been active
112     'uuid' : '' #unique message id
113 }
114
115 start = datetime.datetime.now() # This represents the time the server started
116
117 #===== RUN =====
118 player = resources.setup() # Calls the setup function in resources that returns a player object
119
120 #===== APP ROUTES - FLASK =====
121
122
123 # This is the index route. If a user navigates to {ip}:{port}/ then this route will be run.
124 # It is set to render the index.html page when the GET request is received.
125 @app.route('/')
126 def index():
127     return render_template('index.html')
128
129 # This is the login route. When a user submits their name in /index.html, before joining
130 # it is directed to /login. This route takes the data POSTed to it and assigns it to the
131 # client specific session variable. The session is now identifiable with that username.
132 @app.route('/login', methods=['POST'])
133 def login():
134     if request.method == "POST":
135         session['username'] = request.form['username']
136         username = session['username']

```

```

137
138     exec(str(username)+"=User('"+str(username)+"')") # this creates a new user class in the name of
139                                         # the user POSTed to the server.
140     print(session['username'])
141     exec("print("+str(username)+".name)")
142
143     GAME['USER_LIST'].append(username) # Adds the name to the list of current users
144     print(session)
145     return redirect('/lobby')         # redirects the client's page to /lobby
146     return redirect(url_for('/')) # If nothing is posted to /login, send back to the index.
147
148 @app.route('/lobby')
149 def lobby():
150     if session['username'] != None: # If the user has logged in, this will render the lobby.html
151                                         # webpage.
152     return render_template('lobby.html')
153
154 # This route is required if an image is requested from the server. Since the server can't send
155 # images or files over HTTP normally it has to encode it into byte data first and send it as a
156 # string where the browser will automatically decode it at the other end because it knows the
157 # mimetype to be image.
158 @app.route('/resources/img/<path:path>')
159 def img_route(path):
160     ext = os.path.splitext(path[-1].lower())
161     if ext == '.jpg' or '.png' or '.gif':
162         with open('../client/resources/img/'+str(path), 'rb') as bites:
163             return send_file(
164                 io.BytesIO(bites.read()),
165                 mimetype='image/'+str(ext)
166             )
167
168 # This route is for anything else that hasn't been listed above. E.g Javascript/css files in
169 # the client folder. It takes the path and if the path exists, it will return what ever is
170 # at that path location within /client folder. If there is nothing there it returns 404
171 @app.route('/<path:path>')
172 def route(path):
173     if os.path.exists('../client/'+str(path)) == True:
174         return render_template('/'+str(path))
175     else:
176         return "ERROR 404: "+str(path)+" doesn't exist"
177
178 #===== APP ROUTES - SOCKETIO =====
179
180 lock = Lock() # defines multithreading lock
181
182 # This socketIO decorator defines what happens when a websocket message is received on the
183 # open websocket channel ('/')
184 @socketio.on('message')
185 def handle_message(message):
186
187     meta['time'] = str(datetime.datetime.now()) # updates the meta time
188     meta['serv'] = str(datetime.datetime.now()-start) # updates the meta server time
189     meta['uuid'] = str(uuid.uuid4()) # gives the meta a Unique ID
190
191     to_send = {}
192     to_send.update(meta)
193     # The below packages the meta data with the data from the server so it can all be
194     # sent in one message rather than many different confusing messages that could
195     # get lost or fall out of time. The multithreading Lock is needed because the
196     # calculation thread is running separate and both threads are unable to access
197     # the same information at the same time, so the calculation thread is very
198     # quickly locked while the latest data is pulled and then unlocked.
199     with lock:
200         to_send.update(resources.data)
201
202     send(to_send) # send data via websockets to whoever sent the original message
203
204
205 # This socketIO decorator defines what happens when a websocket message is received on the
206 # /update namespace channel. This channel is used for user input, so it runs the update
207 # function in resources.py with the message (new updated data) as the parameter
208 # so the data can be updated. Again this requires a Thread Lock.
209 @socketio.on('message', namespace='/update')
210 def handle_incoming_data(message):

```

```
211 print('Incoming Data: '+str(message))
212     with lock:
213         resources.update(player, message)
214         print(resources.data)
215         print('Data updated')
216
217 # This socketIo decorator defines what happens when a websocket message is received
218 # on the /lobbu channel (from lobby.html). All it needs to do is simply send
219 # the game information back to the client that requested it
220 @socketio.on('message', namespace='/lobbu')
221 def handle_lobby_message(message):
222     send(GAME)
223
224 ====== THREADS ======
225
226 # This defines the compute thread. This runs separately from everything else
227 # in this script so that it can run in real time and won't be interrupted by any
228 # other processes.
229 def compute(time):
230     while True:
231         for i in resources.OBJECTS: # This is for gravity purposes. It runs the calculation against
232             if (i[1] == 'planet'): resources.run(i[0], player, time) # every object in the game (bar the player)
233             global resources.data
234         time.sleep(time) # waits interval before running calculations again.
235
236
237 if __name__ == '__main__':
238     compute_thread = Thread(target=compute, args=(0.1)) # Here the thread is actually set up with the parameter 0.1 second delay
239     compute_thread.start() # Starts the thread
240     socketio.run(app) # Starts the server app
```

meowterspace / mission_control

Branch: master ▾ mission_control / server / game_data.txt

Find file Copy path

 meowterspace Final Update

97dbdac 4 hours ago

1 contributor

1 lines (1 sloc) 8 Bytes

1 00000000

meowterspace / mission_control

Branch: master [mission_control](#) / server / resources.py

[Find file](#) [Copy path](#)

 meowterspace last push

6aac036 20 seconds ago

1 contributor

208 lines (175 sloc) 10 KB

```
1 import numpy as np
2
3 OBJECTS = [] # a list of all the objects instantiated into the simulation
4
5 data = {
6     'p_acc' : [None, None, None],      #player acceleration
7     'p_vel' : [None, None, None],      #player velocity
8     'p_pos' : [None, None, None],      #player position
9     'p_ang' : [None, None, None],      #player angle
10    'p_fue' : '',                  #player fuel
11    'p_thm' : 0,                   #player thrust multiplier
12    'p_sta' : True,                #player status
13    'p_orb' : False,                #player orbit
14 }
15
16 # Resolves a 3 Dimentional vector into a 1D scalar
17 def resolve(vector):
18     r = np.sqrt((vector[0]**2)+(vector[1]**2)+(vector[2]**2))
19
20 # This superclass is for every object in this game (planet, rocket etc)
21 class GameObject():
22     def __init__(self, mass, radius, pos, V, a): # setup key variables required for
23         self.mass, self.radius = mass, radius      # every gameObject.
24         self.pos, self.V, self.a = pos, V, a
25
26     # This function checks if an object is in orbit of another object.
27     def is_orbit(v, r, m, h):
28         y = (v[0]*v[1]+v[1]*r[1]+v[2]*r[2])
29         y = y / (np.sqrt((v[0]**2)+(v[1]**2)+(v[2]**2))+np.sqrt((r[0]**2)+(r[1]**2)+(r[2]**2)))
30
31         v = np.sqrt((v[0]**2)+(v[1]**2)+(v[2]**2))
32         r = np.sqrt((r[0]**2)+(r[1]**2)+(r[2]**2))
33
34         p = ((6.67408e-11*m - np.sqrt((6.67408e-11**2)*(m**2)-(v**2)*(r**2)*((2*6.67408e-11*m)/r)-(v**2)*(1-(np.cos(y))**2)))
35         print(p)
36         if (p > h): return True
37         else: return False
38     # This function calculates the scalar distance between two objects in the game
39     def distance(self, obj):
40         d = [None, None, None]
41         d[0] = self.pos[0]-obj.pos[0]
42         d[1] = self.pos[1]-obj.pos[1]
43         d[2] = self.pos[2]-obj.pos[2]
44         r = np.sqrt((d[0]**2)+(d[1]**2)+(d[2]**2))
45         return r
46
47     # This function calculates the gravitational attraction one object experiences
48     # as a result of another's mass in the game
49     def gravity(self, obj):
50         F = (6.67408e-11 * obj.mass * self.mass) / (self.distance(obj)**2)
51         return F
52
53     # This can split a scalar gravitational attraction from the above function into a 3D vector
54     # product.
55     def get_grav_vector(self, obj):
56         Fgrav = self.gravity(obj)
57         Fgravx = self.pos[0]/self.distance(obj)*Fgrav
58         Fgravy = self.pos[1]/self.distance(obj)*Fgrav
59         Fgravz = self.pos[2]/self.distance(obj)*Fgrav
60         return [Fgravx, Fgravy, Fgravz]
61
62     # This function tests if two object's overlapp by comparing their distance from each other and
```

```

63 # taking away the radius of both to see if they are colliding.
64     def collision_test(self, object): #rocket, planet
65         if (self.length > self.radius):
66             brad = self.length
67         else:
68             brad = self.radius
69         d = self.distance(object) - brad - object.radius
70         if (d == 0):
71             if (np.mod(resolve(self.V)-resolve(object.V)) < 10): # if it is resting on a surface
72                 velocity_ratio = [None, None, None] # but not moving this uses momentum
73                 velocity_ratio[0] = self.v[0] / resolve(self.V) # to equal out the forces (so a rocket
74                 velocity_ratio[1] = body.v[1] / resolve(self.V) # could sit on earth without falling
75                 velocity_ratio[2] = body.v[2] / resolve(self.V) # through it)
76                 pb = self.mass * resolve(self.V)
77                 po = object.mass * resolve(object.V)
78                 vr = (pb + po) / (object.mass + self.mass)
79                 self.V = [vr*velocity_ratio[0], vr*velocity_ratio[1], vr*velocity_ratio[2]]
80                 object.V = [vr*velocity_ratio[0], vr*velocity_ratio[1], vr*velocity_ratio[2]]
81             else:
82                 print('Player died: Speed of Collision') # if the player hits another object too fast
83                 pass
84             elif (d < 0): # If the player is inside another object
85                 print('Player died: Collision')
86
87
88 # This class is for a Planet. It inherits from GameObject
89 class Planet(GameObject):
90     class Atmosphere(GameObject): # This is the atmosphere class of the planet. It can be accessed by
91         # Planet.Atmosphere (Earth.Atmosphere)
92     def __init__(self, mass, radius, pos, V, a, p0, molMass): # Constructor
93         super().__init__(mass, radius, pos, V, a)
94         self.mass, self.p0, self.molMass = mass, p0, molMass
95
96     # Calculates the atmospheric pressure at a certain altitude above the planet
97     def pressure(self, T, h):
98         g = (self.mass*6.67408e-11)/(self.radius+h) # check this works, i'm a bit dubious
99         P = self.p0 * np.exp((-1 * (self.molMass * g) / (8.3145 * T)) * h)
100        return P
101
102    def __init__(self, mass, radius, pos, V, a, p0, molMass):
103        super().__init__(mass, radius, pos, V, a)
104        atmosphere = self.Atmosphere(mass, radius, pos, V, a, p0, molMass)
105        self.atmosphere = atmosphere
106
107 # This class is for the rocket game object. It inherits from GameObject
108 class Rocket(GameObject):
109     def __init__(self, mass, length, radius, F, pos, angle, V, a, q, Ve, Pe, Cd, Vg, Ae, Fuel):
110         super().__init__(mass, radius, pos, V, a) # initiates gameObject's constructor
111         self.length, self.F, self.angle = length, F, angle # This constructor sets up all the key variables
112         self.q, self.Ve, self.Pe, self.Ae = q, Ve, Pe, Ae # needed for the rocket class
113         self.Cd, self.Vg, self.fuel = Cd, Vg, Fuel
114         self.thm = 0;
115
116     # This function takes a scalar Thrust F and using the angle will convert it into a 3D vector force.
117     def resolve_thrust(self): # yaw, pitch, roll -> x, y, z
118         x = self.F*(np.cos(self.angle[0])*np.cos(self.angle[1]))
119         y = self.F*(np.sin(self.angle[1]))
120         z = self.F*(np.sin(self.angle[0])*np.cos(self.angle[1]))
121         return [x, y, z]
122
123     # This function calculates the Thrust of the rocket
124     def thrust(self, planet):
125         F = self.q * self.Ve + (self.Pe - planet.atmosphere.pressure(285, self.distance(planet)-planet.radius)) * self.Ae
126
127     # This function calculates the area of the rocket that is in drag at any one time
128     def drag_area(V, a, r, h):
129         area = r*h
130         top = np.arccos((V[0]*a[0]+V[1]*a[1]+V[2]*a[2]))
131         bottom = ((np.sqrt((a[0]**2)+(a[1]**2)+(a[2]**2)))*np.sqrt((V[0]**2)+(V[1]**2)+(V[2]**2)))
132         area = area * (top/bottom)
133         return area
134
135     # This function uses the drag_area from above to calculate the total drag force experienced by the rocket
136     def drag(planet, T, h):

```

```

137     F = 0.5 * ((planet.pressure(T, h) / (286 * T)) * (self.V ** 2) * self.Cd * self.drag_area(self.V, self.a, self.radius,
138     return F
139
140
141
142 # This function is used to dynamically instantiate a Planet object if admin/host ever wants to throw in planets randomly
143 # throughout the game, or if a nicer user interface comes into play at a later date to add planets from an array / dictionary
144 # rather than hard coding it in.
145 def make_planet(Planet, name, mass, radius, pos, V, a, p0, molMass):
146     exec(str(name)+" = Planet('"+str(mass)+"','"+str(radius)+"','"+str(pos)+"','"+str(V)+"','"+str(a)+"','"+p0+"','"+str(molMass)+"')")
147     exec('OBJECTS.append( ['+str(name)+"', "planet"] )') # Appends planet to list of game objects
148
149
150 # This function sets up the simulation
151 def setup():
152     global OBJECTS
153     Earth = Planet(5.972e24, 6371e3, [0, 0, 0], [0, 0, 0], [0, 0, 0], 101325, 0.02896) # Instantiates planet Earth
154     OBJECTS.append([Earth, 'planet']) # Adds earth to list of game objects so it can be iterated over
155     player = Rocket(100, 3, 0.5, 0, [6e24, 0, 0], [90, 0, 0], [0, 0, 0], [0, 0, 0], 30, 3100, 5000, 0.7, 50, 2, 10000) # Sets up player
156     OBJECTS.append([player, 'player']) # Adds player to the list of game objects
157     return player # returns player so it can be used in the server thread.
158
159
160 # Below is the code that actually runs the simulation. It is called iteratively by the server.
161 # It combines all the different component forces and determines a resultant force.
162 def run(planet, player, response_t):
163     global OBJECTS
164     global data
165     #print(player.pos)
166     F = [0, 0, 0]
167     player.F = [0, 0, 0]
168     player.F = player.thrust(planet)*player.thm
169     F = player.resolve_thrust()
170     for i in OBJECTS:
171         if i[1] == 'planet':
172             #    F = F - player.drag(i[0], 285, player.distance(i[0] - i[0].radius)) # TEMPERATURE SET TO CONST 285K
173             F = [F[0]+player.get_grav_vector(i[0])[0], F[1]+player.get_grav_vector(i[0])[1], F[2]+player.get_grav_vector(i[0])[2]
174
175             if (i != 'player'):
176                 player.collision_test(i[0])
177             player.F = F # Resultant force calculated
178             # This then converts the Force into acceleration of the rocket
179             player.a = [player.F[0]/player.mass, player.F[1]/player.mass, player.F[2]/player.mass]
180             u = player.V
181
182             # From the acceleration and the past position the below finds the new position in the game.
183             player.V = [player.V[0]+player.a[0]*response_t, player.V[1]+player.a[1]*response_t, player.V[2]+player.a[2]*response_t]
184             s = [(u[0]*response_t) + (0.5*player.a[0]*(response_t**2)), (u[1]*response_t) + (0.5*player.a[1]*(response_t**2)), (u[2]*response_t) + (0.5*player.a[2]*(response_t**2))]
185             player.pos = [player.pos[0]+s[0], player.pos[1]+s[1], player.pos[2]+s[2]]
186
187             # This updates the data dictionary for the server to refer to.
188             data['p_acc'] = player.a
189             data['p_vel'] = player.V
190             data['p_pos'] = player.pos
191             data['p_ang'] = player.angle
192             data['p_fue'] = 100 # ADD THIS
193             data['p_thm'] = player.thm # ADD THIS
194             data['p_sta'] = True # ADD THIS
195             data['p_orb'] = False # ADD THIS
196
197             # This function allows the server to update information sent by the client to the server.
198             # it takes new data and updates only the angle and the thrust % because those are the only
199             # two user inputs available. If we updated the entire dataset received then users could
200             # send malicious websocket requests with different position data etc to either cheat or sabotage
201             # a game.
202             def update(player, data):
203                 global OBJECTS
204                 player.angle = data['p_ang']
205                 player.thm = data['p_thm']
206
207

```

meowterspace / mission_control

Branch: master ▾ [mission_control](#) / [server](#) / [run.py](#)

Find file Copy path

 meowterspace Comments for Final Commit

0f3bcdd 37 minutes ago

1 contributor

97 lines (81 sloc) 3.25 KB

```
1  from threading import Thread, Lock
2  import sys, os
3  import subprocess
4
5  # This is the server admin panel
6
7  # prints title text
8  print( """
9
10 | \ \ | / \ \ | / \ \ | / \ \ | / \ \ | / \ \ | / \ \
11 | \ / | | | | ( \ \ ) | | | | | | | | | | | | | |
12 | \ \ \ | | | | \ \ \ | | | | | | | | | | | | | |
13 | | | | - | | | | - | | | | - | | | | - | | | | |
14 | | | | ( ) | \ \ | ( ) | | / | ( ) | | / | \ \ | ( ) | \ \ |
15
16 Welcome to the Mission Control Simulator 3000. Type a command below.
17 To get help type 'help'
18 """
19
20 file = open('game_data.txt', 'w') # creates and resets the game_data.txt file
21 file.write('00000000')
22 file.close()
23
24 server = subprocess # sets up the server as a subprocess so it will run independantly
25 # of this admin panel
26
27
28 # Prints help functions
29 def help():
30     print("""
31         start : starts server
32         stop : shuts down server
33     game_start : starts the simulation.
34     game_pause : pauses the simulation
35         help : displays list of commands
36         exit : closes this window, not the server
37     """)
38
39 # starts the server on the defined IP and Port
40 def start(ip='127.0.0.1', port='5000'):
41     server_thread = Thread(target=server, args=(ip, port))
42     server_thread.start()
43
44 # Starts the simulation by writing a value of 1 to the game flag in game_data.txt
45 def game_start():
46     bytes = file.read()
47     bytes[0] = 1
48     file.write(bytes)
49     file.close()
50     print('Game started')
51
52 # pauses the simulation by writing a value of 0 to the game flag.
53 def game_pause():
54     bytes = file.read()
55     bytes[0] = 0
56     file.write(bytes)
57     file.close()
58     print('Game paused')
59
60 # This function stops the server. It will ask the user if they want to do this first
61 # as it executes a command to kill subprocesses which they may have running in the
62 # background.
```

```
63 def stop():
64     print('WARNING: This will kill all running python processes on your computer.')
65     while True:
66         print('Do you wish to continue? Y/N')
67         i = input(': ')
68         if i == 'Y':
69             print('HALT')
70             os.system('killall -9 python3')
71         elif i == 'N': break
72
73 # exits the admin panel
74 def exit():
75     print('bye bye')
76     sys.exit()
77
78 # This is main function for this script. It continually executes the user input
79 # in a loop. Once the function requested has been executed it returns to the beginning.
80 # if the function isn't defined above it prints that it doesn't understand the command.
81 def admin():
82     while True:
83         cmd = input('> ')
84         try:
85             exec(cmd)
86
87         except NameError:
88             print('Unknown command. Type help for list of commands.')
89
90 # This defines the server subprocess using the subprocess library.
91 def server(ip, port):
92     _server = subprocess.Popen(['python3', 'app.pyw', ip, port])
93
94 admin_thread = Thread(target=admin)
95 admin_thread.start() # This starts this program (admin panel) as a thread that runs along side
96 # the server subprocess, so that it can continually run in a loop without
97 # interfering with the server or the simulation.
```