Virtual Mission Control for Aerospace Applications

VMCAA

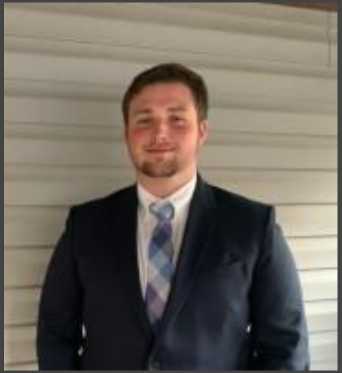
Preliminary Design Review

Matt Dow, Carson Watkins, John Stanfield

4/22/21

Team Members

Matt Dow
Frontend / Data Visualization



John Stanfield
Backend / Simulation

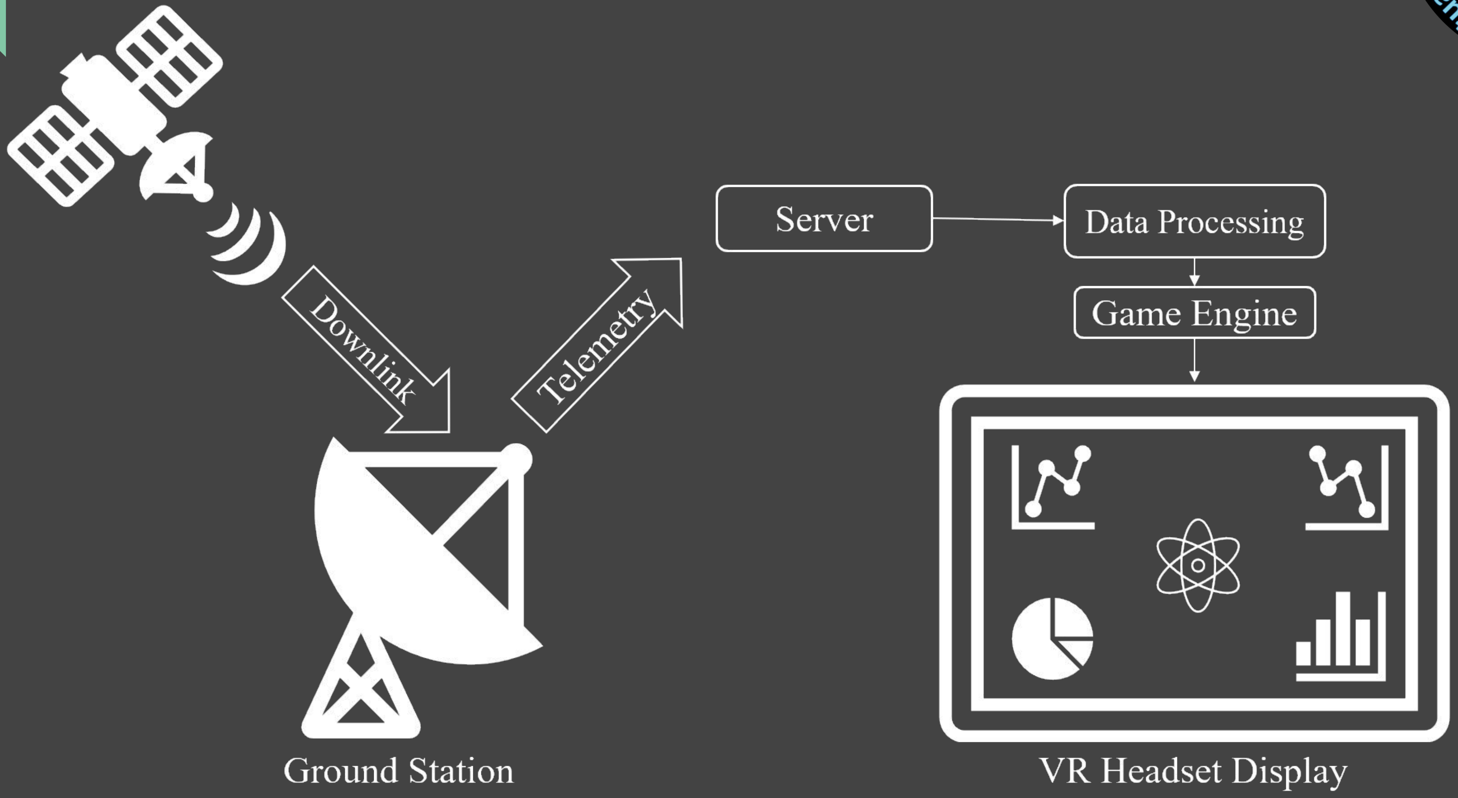
Carson Watkins
User Interaction / Environment

System Overview

- Make satellite management cost effective and portable for smaller companies and universities.

- Build a system capable of visualizing and monitoring satellite constellations using a Virtual Reality system.

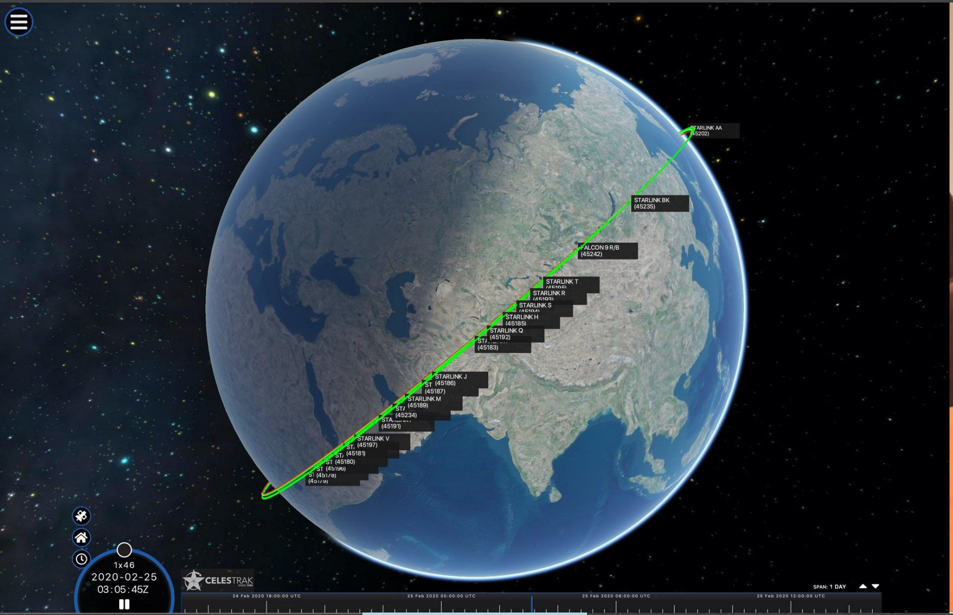
Concept of Operations

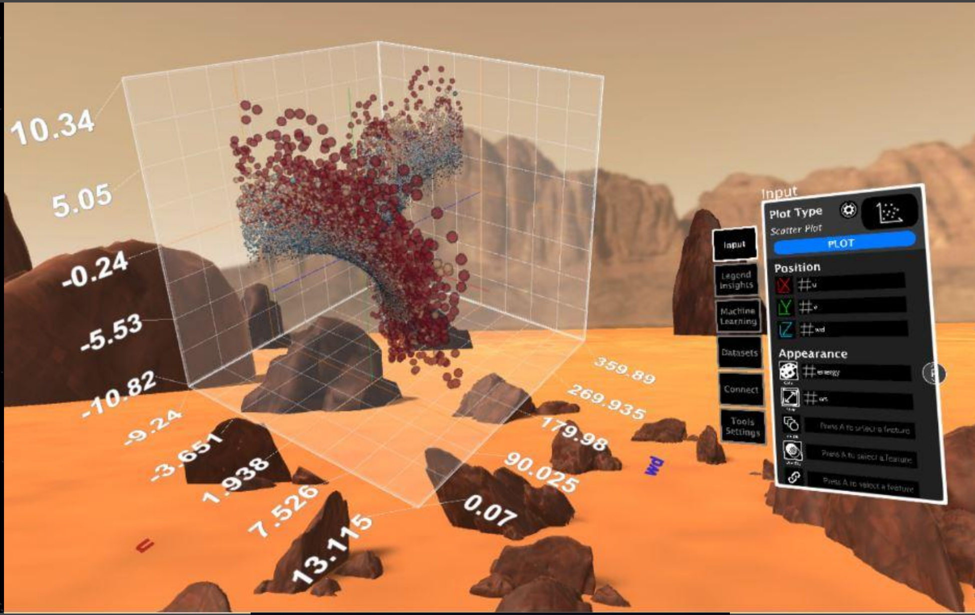


Similar Systems

VMCAA

Virtual Mission Control for Aerospace Applications

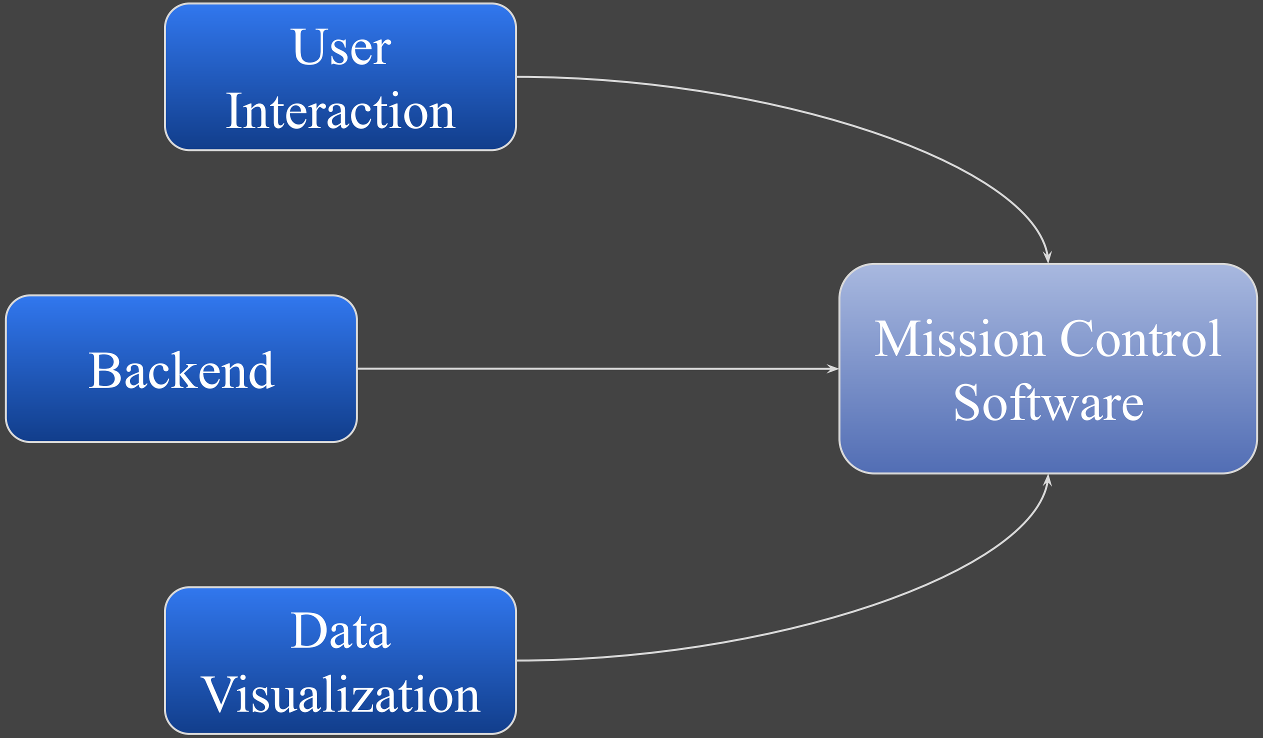




CelesTrak

VIRTUALITICS

System Description



System Demonstration

Failed Requirements

Requirement Type	#	Requirement Statement	Status
Functional	1.4.1	The software shall allow for satellite anomalies in physical constraints to be detected.	🟥
Performance	2.1.2	The system shall display each satellite's temperature and power data.	🟥
Performance	2.4.1	The system shall display at least 1262 separate orbits.	🟨

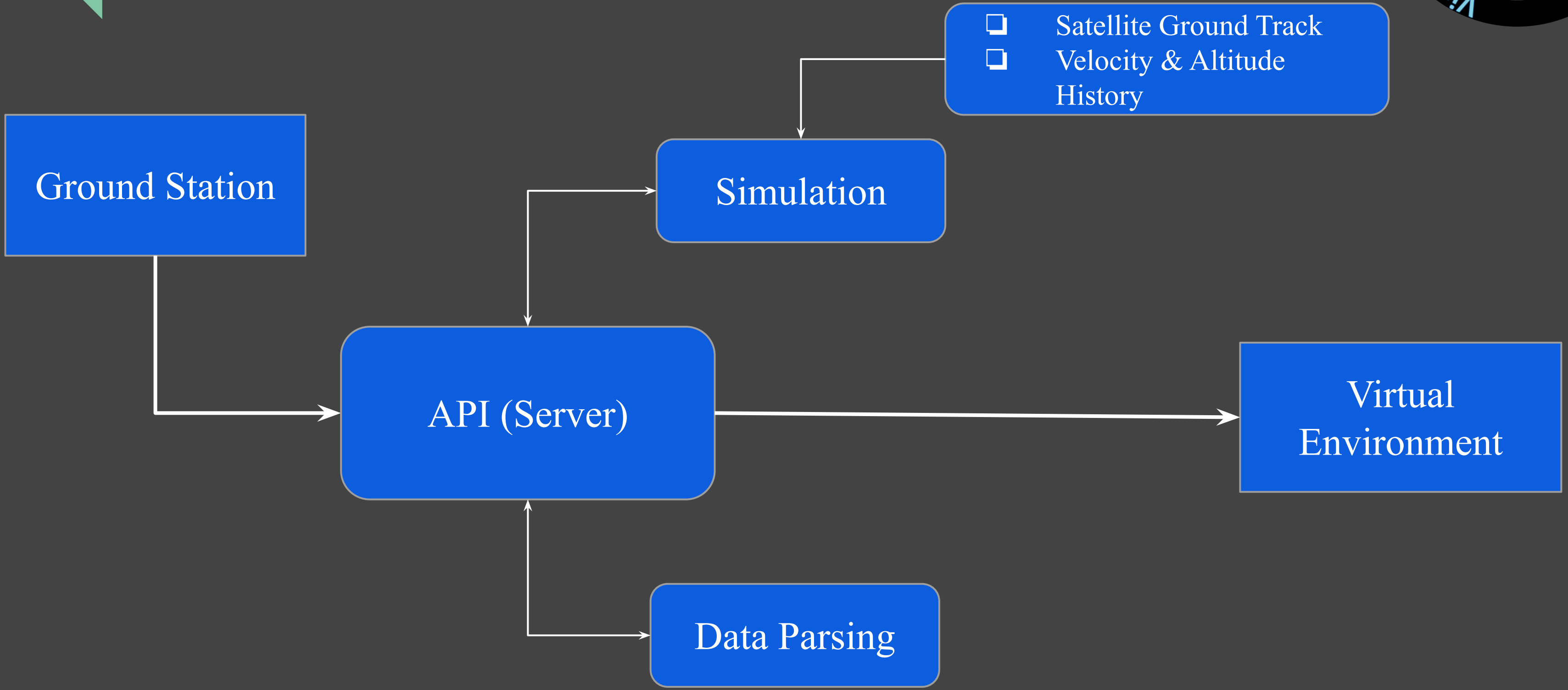
Consumer Budget

Virtual Mission Control Budget	
Required Items	Cost
Oculus Virtual Reality Headset	$300 - 500
Server (Two Options) ● Dell Server (Built/Configured Server) ● Dedicated PC (only recommended for small uses)	Purchased Server : $500 - $2000 Dedicated PC: $0 - $2000
Total Cost	Minimum: $300 Maximum: $2500

Jay Stanfield

Backend / Simulation

Subsystem Overview



❑ Satellite Ground Track
❑ Velocity & Altitude History

Design Drivers

Requirement Type	#	Requirement Statement
Functional	1.2	The system shall allow for data transfer between the analysis and visualization platforms.
Functional	1.3	The system shall allow for simulation of orbital data.
Performance	2.1	The system shall receive TLEs from a ground station.
Performance	2.4	The system shall automatically update data.

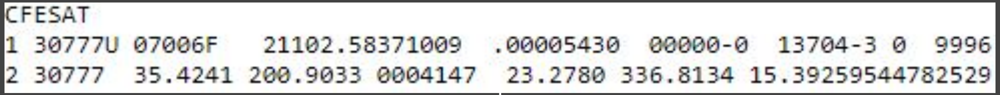
API Design

Data Transfer via RESTful API				
Method	Effectiveness	Maintenance	Ease of Use	Total
Python/Django Framework	9	4	2	15
Javascript/Express Framework	9	6	3	18

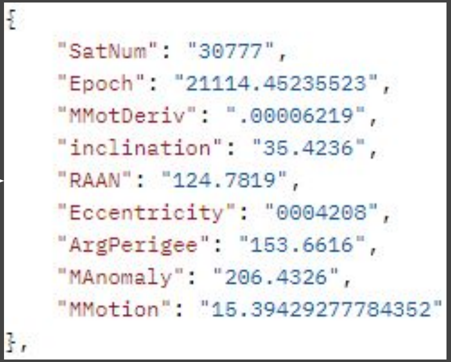
```
app.get('/api/grndtrk', (req, res) => {
    const spawn = require("child_process").spawn;

    const pythonProcess =
spawn('python',["simulation.py"]);
```

Telemetry



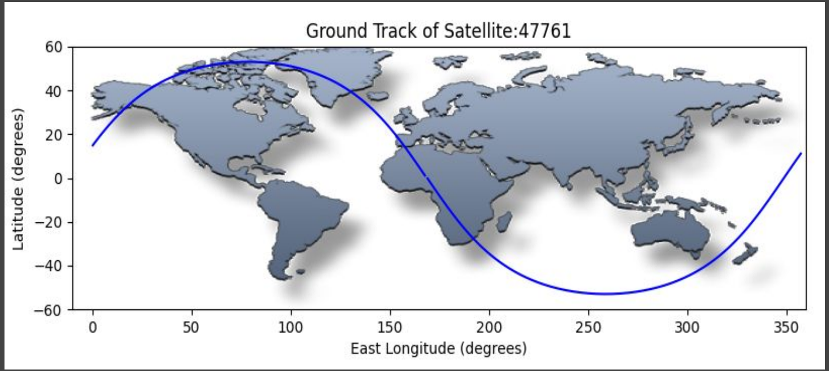
```
CFESAT
1 30777U 07006F   21102.58371009  .00005430  00000-0  13704-3 0  9996
2 30777  35.4241 200.9033 0004147  23.2780 336.8134 15.39259544782529
```



```
{
    "SatNum": "30777",
    "Epoch": "21114.45235523",
    "MMotDeriv": ".00006219",
    "inclination": "35.4236",
    "RAAN": "124.7819",
    "Eccentricity": "0004208",
    "ArgPerigee": "153.6616",
    "MAnomaly": "206.4326",
    "MMotion": "15.39429277784352"
},
```

Simulation Design

- Model is based on:
  - Two-Body Dynamics
  - J2 Perturbations
- Numerical Simulation:
  - Runge Kutta 4th Order Integrator
- Interactivity
  - Allows for the user to submit a satellite ID for analysis.
- Ground Track



Subsystem Analysis

Backend Performance	
System	Response Time (seconds)
Simulation	0.40625
TLE Update	0.046875

Subsystem Requirements Status

Requirement Type	#	Requirement Statement	Status
Functional	1.2	The system shall allow for data transfer between the analysis and visualization platforms.	🟩
Functional	1.3	The system shall allow for simulation of orbital data.	🟩
Performance	2.1	The system shall receive TLEs from a ground station.	🟩
Performance	2.4	The system shall automatically update data.	🟩

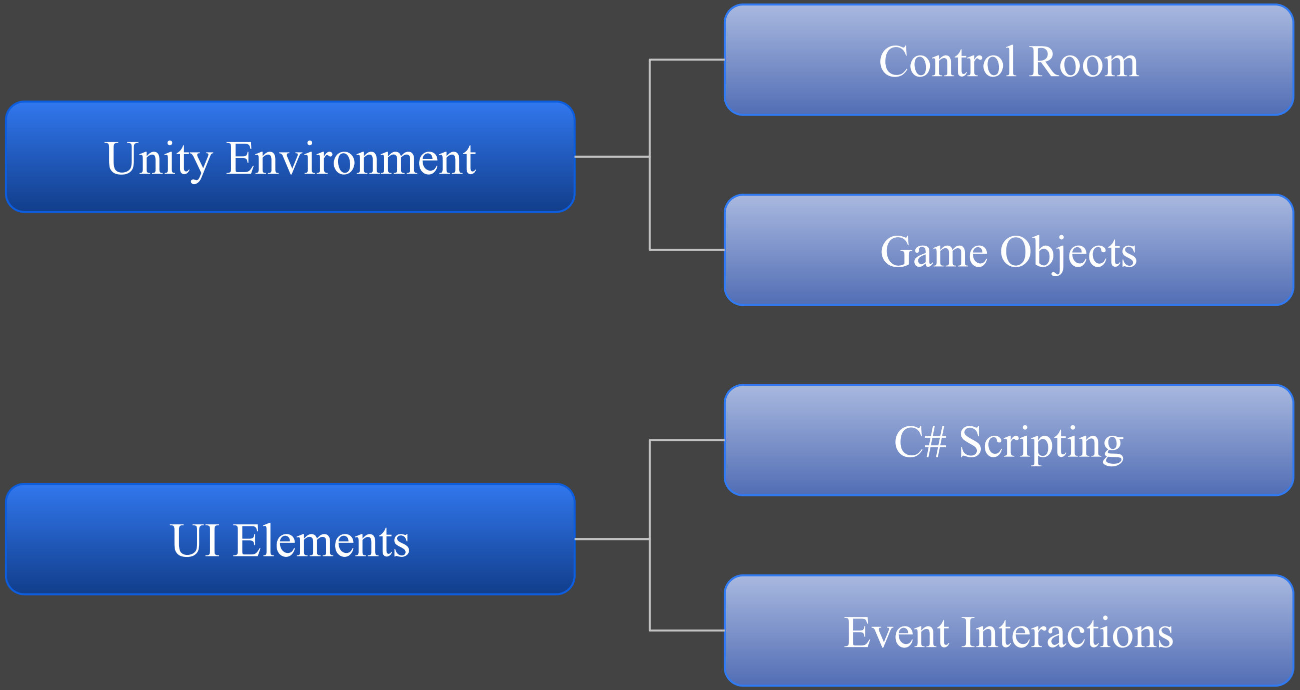
Carson Watkins

User Interaction/Environment

Design Drivers

Requirement Type	#	Requirement Statement
Functional	1.1	The system shall allow for a user to interact with the virtual environment.
Performance	2.3	The system shall allow for the user to interact with satellite data.
Constraint	3.3	The system shall offer a health conducive environment.
Constraint	3.3.1	The system shall reduce movements that could cause motion sickness.
Constraint	3.3.2	The environment shall utilize lighting that reduces strain on the eyes.

Subsystem Overview



Unity Environment

- Control Room
- Game Objects

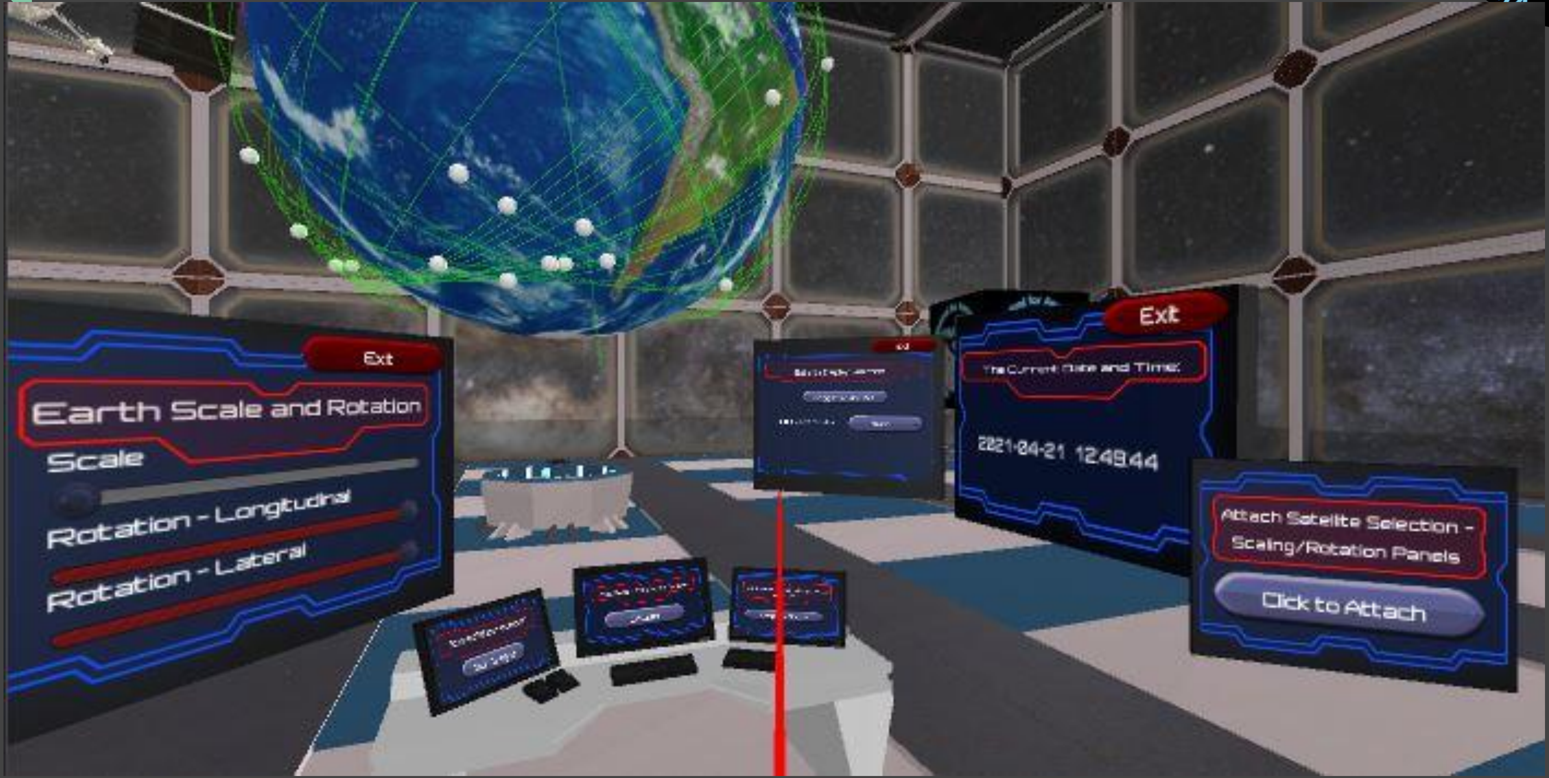
UI Elements

- C# Scripting
- Event Interactions

Unity XR vs. OVR

UI Weighted Values					
Package	Ease of Construction	Environment Interaction	Prefabs	Implementation in Unity	Total
XR	**6**	**12**	**2**	**9**	**29**
OVR	**6**	**4**	**3**	**3**	**16**

Environment and Interaction



The Current Date and Time:

2021-04-21 12:49:44

Requirement Status

Requirement Type	#	Requirement Statement	Status
Functional	1.1	The system shall allow for a user to interact with the virtual environment.	🟩
Performance	2.3	The system shall allow for the user to interact with satellite data.	🟩
Constraint	3.3	The system shall offer a health conducive environment.	🟩
Constraint	3.3.1	The system shall reduce movements that could cause motion sickness.	🟩
Constraint	3.3.2	The environment shall utilize lighting that reduces strain on the eyes.	🟩

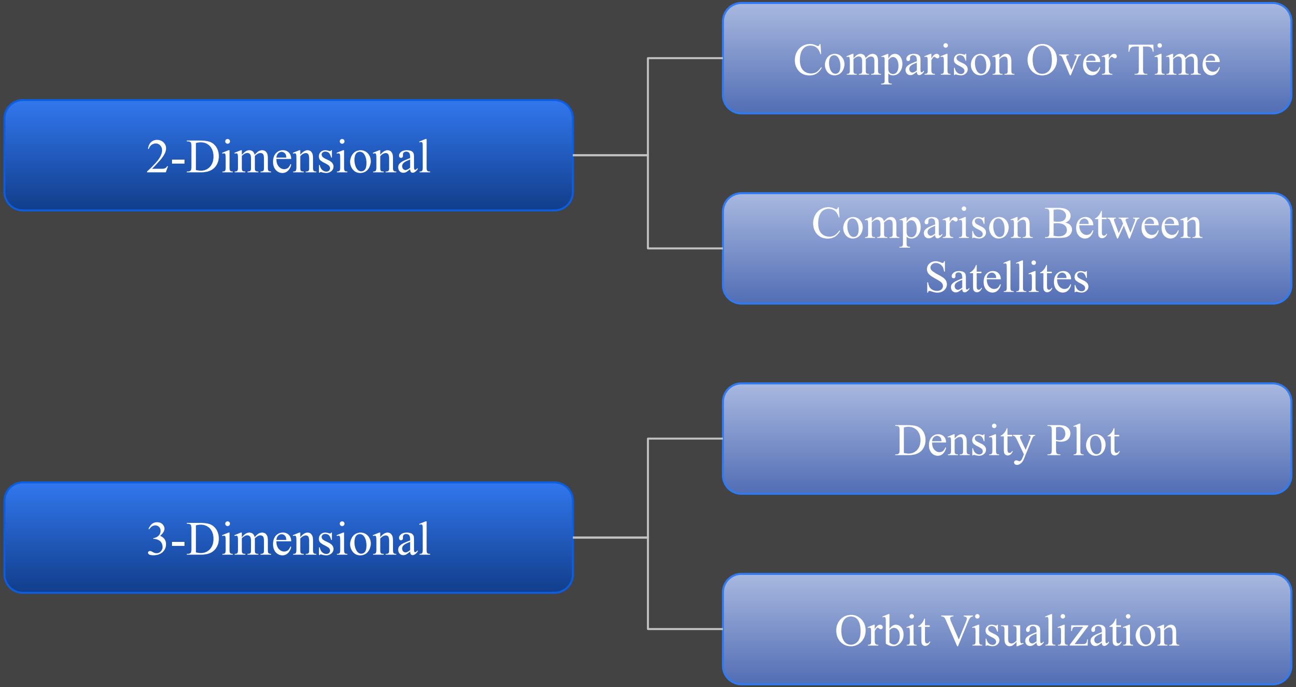
Matt Dow

Data Visualization

Design Drivers

Requirement Type	#	Requirement Statement
Performance	2.1	The system shall receive and visualize data from TLEs.
Performance	2.2	The system shall allow for visualization of multiple satellite constellations.
Performance	2.4	The system shall update visualized data autonomously.

Subsystem Overview



2-Dimensional

- Comparison Over Time
- Comparison Between Satellites

3-Dimensional

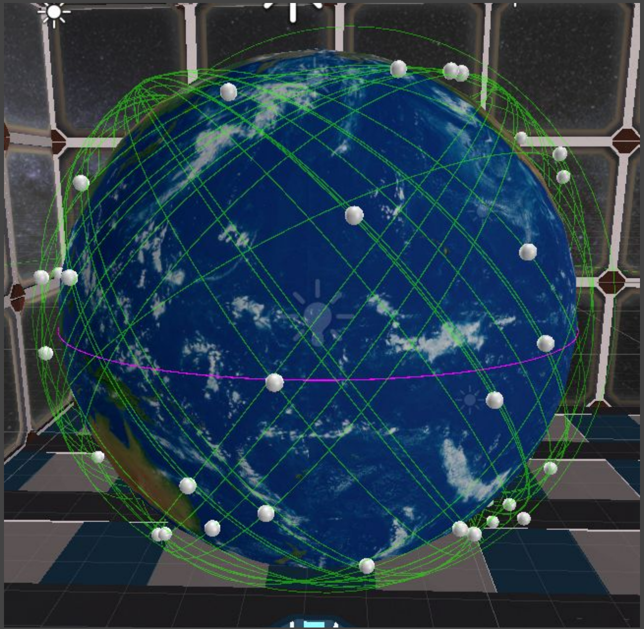
- Density Plot
- Orbit Visualization

Trade Study Outcome

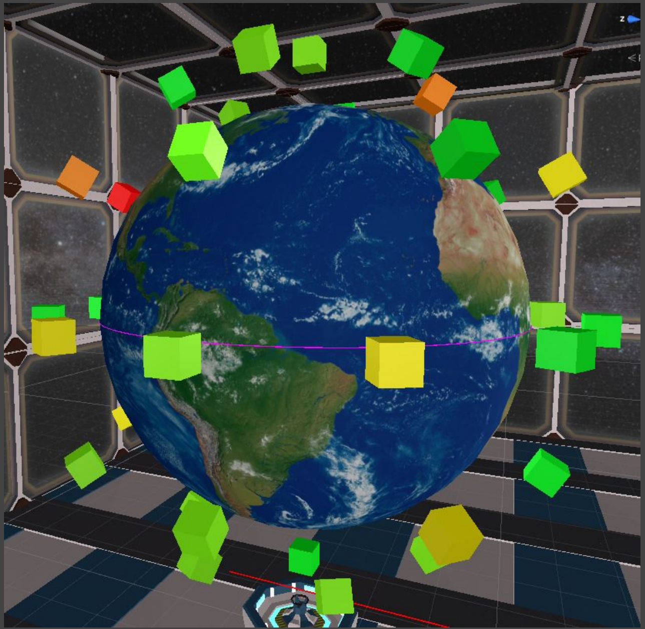
Note: All graph types were implemented into the system, this trade study only affected the order in which each type was prioritized.

UI Weighted Values					
Package	Amount Displayable	Readability	Processing Load	Time Span	Total
Bar Graph	**9**	**12**	**6**	**1**	**28**
Histogram (Individual)	**3**	**12**	**6**	**2**	**23**
Dot Graph (Group)	**6**	**8**	**4**	**2**	**20**

3D Data Visualization

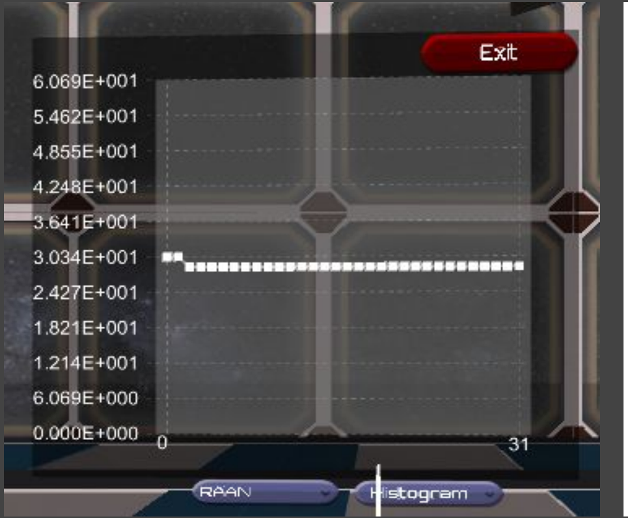


Orbit Visualization

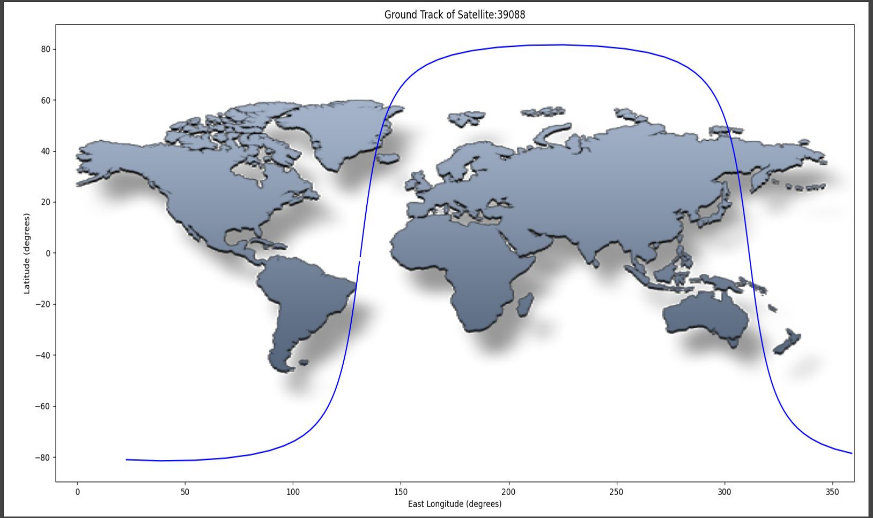


Satellite Density

2D Data Visualization (Time History Comparison)

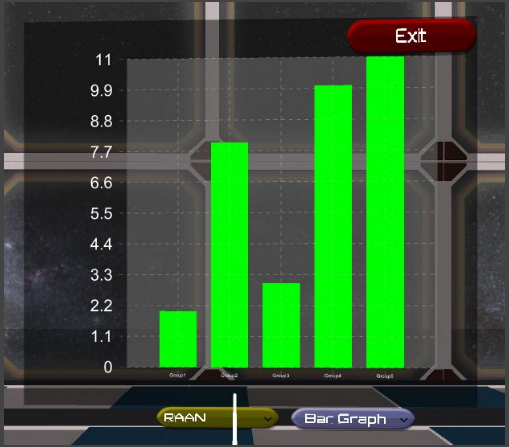


Element History

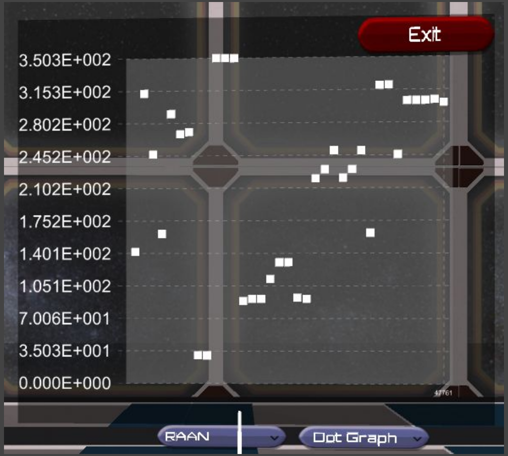


Ground Track

2D Data Visualization (Constellation Comparison)



Comparison of Data Ranges



Comparison Of Individual Data vs Others

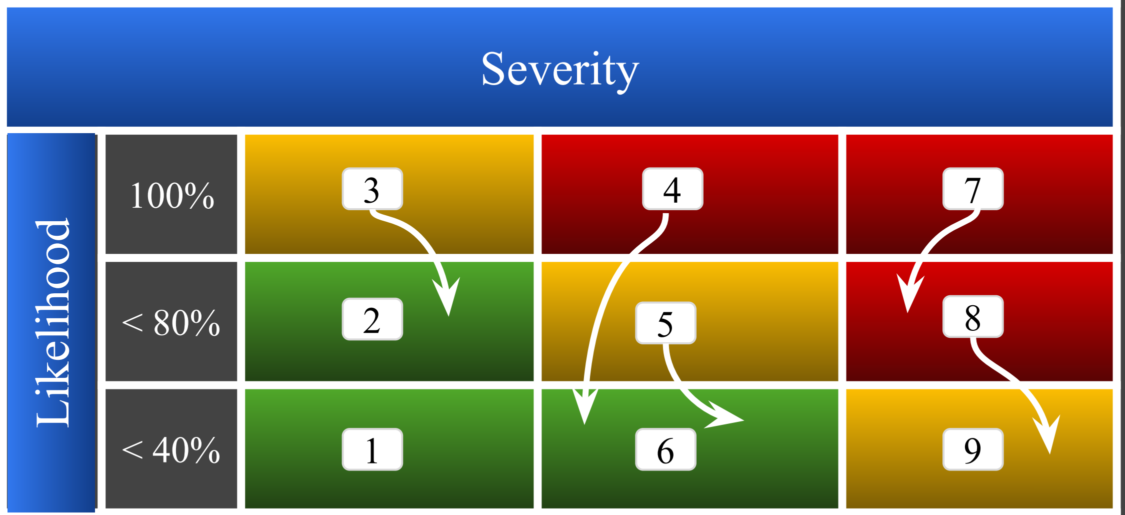
Requirement Status

Requirement Type	#	Requirement Statement	Status
Performance	2.1	The system shall receive and visualize data from TLEs.	🟩
Performance	2.2	The system shall allow for visualization of multiple satellite constellations.	🟩
Performance	2.4	The system shall update visualized data autonomously.	🟩

Risk Assessment

VMCAA Risk

1. Platform Requires Modification
2. Software Integration Issues
3. Equipment For Development Malfunctions
4. Light/Dizziness Side Effects
5. Data Feed Is Interrupted
6. Software Design Is Too Complex
7. Platform Is Not Capable Of Running Software
8. Injury Due To Side Effects
9. System Compromised Externally



		Severity		
Likelihood	100%	3	4	7
	< 80%	2	5	8
	< 40%	1	6	9

Final Thoughts

Lessons Learned

- Integrate Subsystems Earlier
- Focus main effort on API development during initial phases
- Ensure all group members can run virtual reality systems on personal computers

Future Work

- Implement anomaly tracking system
- Display ground track of a selected satellite
- Implement satellite selection from visualized orbits