**PROJECT: SPACE SITUATIONAL AWARENESS**

1. **PROBLEM STATEMENT:**

Space has become a critical domain for both civilian and military activities. Due to the growing popularity of space exploration among nations, the number of satellites in space is constantly increasing.

Hundreds of objects entered orbit in the last few months, primarily launched by the United States and China. The vast majority were private satellites deployed by SpaceX for its Starlink constellation.

Number of objects launched

* **Total launches:** As of late August 2025, there have been 113 orbital launches from the U.S. and 50 from China, according to data from Gunter's Space Page.
* **Total objects:** A single launch can deploy dozens of satellites. SpaceX launched 9 Falcon 9 missions in May 2025, deploying around 250 satellites for the Starlink constellation. This rate has continued throughout 2025.

This means with the increasing number of satellites comes the growing number of space debris. Debris is made up of defunct satellites, spent rocket stages, and other fragments that are no longer operational but still orbiting the Earth.

**RISK:** **This debris poses a significant risk to operational satellites and spacecraft. Collison with micrometeoroids or other orbital debris can damage a satellite and end its mission hastily**.

1. **PROPOSED SOLUTION**

**BUILD A COLLISON RISK AND MANAGEMENT SYSTEM**

**Dataset Context**: **ELSET**

The Element set dataset contains Two -Line sets (TLEs) which only provide information on orbital parameters. TLEs can help compute position vectors (x,y,z) and velocity vectors (vx,vy,vz) and Orbital elements (semi-major axis, eccentricity, inclination, RAAN, argument of perigee, true anomaly). This information will help predict orbits but is limited in scope.

To create a robust system, it is recommended to explore other factors including:

* Object Size or Mass - This affects collision damage risk.
* Cross sectional area - Important for drag and probability of collision
* Radar/Optical cross -section (RCS) – This is used to estimate object detectability & Size
* Material type – Helps with survivability estimates after impact
* Atmospheric drag models – For Low Earth Orbits (LEO) objects which tend to have shorter lifespans due to atmospheric drag.
* Space weather (solar activity) which affects orbital decay
* Historical collision events and fragmentation models help with probabilistic risk.

This means reviewing additional dataset provided by UDL or integrate other external data sources.

1. **CORE ANALYSIS**

The system should be able to do:

1. **Collision Probability Estimation**

* Use relative position + velocity between objects.
* Methods: *Conjunction Analysis (CA)*, *Poisson models*, or Monte Carlo simulations.

1. **Close Approach Prediction**

* Propagate orbits forward in time → check for objects entering a threshold “safety bubble” (e.g., 5–10 km).

3. **Debris Cloud Simulation**

* If collision occurs, simulate debris distribution using NASA breakup models.
  + 1. **Risk Prioritization**
* Combine collision probability + object importance (e.g., ISS vs. small CubeSat).
* Output a “risk index” for decision-making.
  + 1. **Mitigation Strategies**
* Predict when debris will re-enter atmosphere.
* Suggest orbit adjustments (if propulsion available).
* Highlight safe zones / graveyard orbits.

1. **SYSTEM ROADMAP**
2. Data pipeline

* Integrate TLEs + Auxiliary dataset containing (Object size, drag, space weather)
* Clean and standardize the data.

1. Orbit Propagation

* Use SGP4 (which is how our ELSET data was generated.)
* Extending with higher precision models (E.g., J2 Perturbations, drag)

1. Collision Detection Engine

* Pairwise distance calculation.
* Conjunction analysis Eg find objects with minimum miss distances
* Probability of collision modeling

1. Visualization & Dashboard

* 3D Orbit plots
* Heatmaps of high-risk regions (LEO, GEO (Geostationary orbit)
* Alerts for close approaches

1. Debris Management Layer

* Suggest avoidance maneuvers
* Prioritize which satellites to save
* Model debris evolution

1. **TOOLS TO USE**

Python Libraries

* sgp4 → orbit propagation from TLEs
* poliastro → orbital mechanics analysis
* astropy → time & coordinate systems
* pandas, numpy, scipy → analysis
* matplotlib, plotly → visualization

Data Storage

* Google BigQuery (only if datasets are huge)
* PostgreSQL with PostGIS (for spatial queries)

Dashboard

* Streamlit / Plotly Dash for interactive UI.

1. **END GOAL: RISK COLLISON AND DEBRIS MANAGEMENT SYSTEM**

* A system that continuously ingests orbital + debris data
* Predicts high - risk collisions
* Simulates debris clouds &
* Suggests mitigation actions for space agencies and operators.