🚀 Project 01: Missile Geometry 101

World Defense Organization (WDO)

You have been

🧠 Scenario

You are a **Spatial Defense Analyst** working for the **World Defense Organization**.

Earth is under threat from **non-human entities**:

- Alien spacecraft
- Orbital kinetic weapons
- High-altitude airborne platforms
- Kaiju-class ground threats (yes, really)

Your mission is **not** to fire weapons.

Your mission is to **analyze geometry**.

Specifically:

- Where threats are coming from
- Where they are going
- What they intersect
- Who (or what) might be affected

This project is about **spatial reasoning**, not realism.

🎯 Learning Objectives

By the end of this project, you will be able to:

- Represent spatial features as **points, lines, and polygons**
- Compute **future locations** using bearing and distance
- Build **trajectories** from simulated movement
- Determine **spatial relationships** (within, intersects, distance)
- Verify results using **interactive maps**

If you can *see* it, you can *trust* it.

🧰 Tools & Libraries

You are expected to use:

- **Python**
- **GeoPandas**

- **Shapely**
- **Folium**

No JavaScript frameworks.
No databases.
No real-time simulation.

This is **Missile Geometry 101**, not Star Wars engineering.

🏰 Your Base (Fixed Location)

Each student is assigned **one base location** (lat/lon).

- This is your **command center**
- You do not move
- You defend a large surrounding region

You are responsible for analyzing **multiple incoming threats**.

🚀 Incoming Threats (Simulated)

Threats are generated automatically using provided starter code.

Each threat has:

- `origin_lat`
- `origin_lon`
- `bearing` (degrees)
- `speed` (km per hour)
- `launch_time`
- `threat_type`
    - `"alien"`
    - `"orbital"`
    - `"airborne"`
    - `"kaiju"`

You **do not** generate threats.
You **analyze** them.

📍 Milestone 1 — Plot the World

Goal: Prove you can load and visualize spatial data.

Tasks

- Load a world countries shapefile (or GeoJSON)
- Display it using Folium
- Add your base location as a point marker

Checkpoint Questions

- Can you clearly identify your base on the map?
- Does zooming and panning work correctly?

📸 **Screenshot required**

📏 Milestone 2 — Distance & Bearing

Goal: Reason about spatial relationships numerically *and* visually.

Tasks

- Compute the distance from each threat origin to your base
- Identify the **closest threat**
- Display threat origins as points

Concepts Reinforced

- Haversine distance
- Units (degrees vs kilometers)
- Attribute inspection

📸 **Screenshot required**

➖ Milestone 3 — Trajectories (Point → Line)

Goal: Turn motion into geometry.

Tasks

- For each threat:
  - Compute a **destination point** after a fixed time interval
  - Generate intermediate points
  - Construct a `LineString` trajectory
- Plot trajectories on the map

Visual Expectation

You should clearly see:

- Where threats started
- Where they are headed
- How paths differ by bearing and speed

📸 **Screenshot required**

🛖 Milestone 4 — Intersections & Borders

Goal: Determine what the threats interact with.

Tasks

- Determine:
  - Which country polygons each trajectory **intersects**
  - Whether a trajectory passes **within a threshold distance** of your base
- Highlight intersected countries on the map

Spatial Relationships Used

- `intersects`
- `within`
- distance thresholds

📸 **Screenshot required**

💥 Milestone 5 — Damage Zones (The Bridge)

Goal: Prepare data for the next project.

Tasks

- Create a **buffer zone** around each trajectory endpoint
- Buffer size depends on `threat_type`
- Determine which countries fall within damage zones

Output

A table like:

country	threat_type	severity

This dataset will be reused in **Project 02**.

📸 **Screenshot required**

🎨 Visualization Requirements (Non-Negotiable)

Your final map must include:

- World boundaries
- Base location
- Threat origins
- Trajectories
- Damage buffers (semi-transparent)
- At least one legend or clear visual explanation

If I can't understand your analysis by *looking*, it's not done.

🚀 Stretch Goals (Optional but Dangerous)

Choose **one**:

- Animate threat movement using time steps
- Color trajectories by threat type
- Identify **first country impacted** per threat
- Add altitude metadata (visualized symbolically)

📦 What You Turn In

- In your "completed assignments" folder create a subfolder called `Project_01`.
- It will include a README that professionally describes and organizes the items you turn in.
    - It should include any iPython notebooks or Python scripts you wrote while working on your project[^1].
    - Your generated maps should be included. Screen shots are required for your repo, but we can discuss where you might can host your mapping solutions so they remain available for a while. -**Screenshots embedded in README** (so you don't forget)
    - Your Thoughts. And rememember, trying something and failing is a good topic to include. It makes for an interesting project summary or as part of a presentation. (Everyone likes to know that they weren't the only ones to think of a wrong solution)
        - What surprised you?
        - What broke?
        - What suddenly “clicked”?
    - COMMENTING CODE
    - README FILES

🧪 Grading Rubric (Condensed)

Category	Points
Correct geometry usage	30
Spatial relationships	25
Visualization clarity	25
Code organization	10
Reflection quality	10
Total	**100**

🧠 Final Thought

This project is not about missiles.

It’s about **thinking spatially**:

- Showing Motion (without animation)
- Interaction (without a gui)
- Consequences (if the WDO fails)

Everything else this semester builds on this.

Footnotes

[^1]: Even if you have files that you end up not working with, or code that took you away down a wrong path, please include it. This is a project over time. It should have issues, and not all your decisions will be amazing.