Sensor Coverage Simulation Code Version Tracking:

Version 0.1:

Initial development of sensor.py, target.py, environment.py, and env\_test.py

Sensors and targets are successfully generated.

Sensors track targets as they enter their fov and until the exit.

Battery level deplete as the sensors are active.

Env\_test displays targets that were tracked.

1. Sensor.py
   1. Developed the base class
   2. Functions
      1. draw sensors
      2. update\_sensor\_fov
      3. update\_energy
2. target.py
   1. Developed the base class
   2. Functions
      1. Draw\_target
      2. Move
3. Environment.py
   1. Developed base class
   2. Functions
      1. Create\_env
      2. Generate\_target\_list
      3. Run\_env
4. Env\_testing.py
   1. Initial testing program

Next efforts:

Generate a batch of sensors based on locations and initial fov angle.

Implement the dt to limit framerate?

Create a global environment var for tracking current energy level

Develop a function to calc and capture all stats from the environment instance for later analysis?

Energy

Targets tracked

Total targets

Coverage percent per time tick?

Reset function for sensors that go inactive -reset their initial angle

Add id to sensors