System for High Integrity Elimination of Low-Altitude Drones (SHIELD)

First Semester Project Plan

1. Project Title

System for High Integrity Elimination of Low-Altitude Drones (SHIELD)

2. Team Members

- Bryson Noble (SWE) - bnoble2023@my.fit.edu
- Sophia Versage (AEE) - sversage2024@my.fit.edu
- Edward Ledwith (AEE) - eledwith2023@my.fit.edu
- Nick Jacobson (AEE) - njacobson2023@my.fit.edu
- Aryaa Vijay (AEE) - avijay2024@my.fit.edu
- Miguel Mohammed-Mueses (AEE) - mmohammedmue2023@my.fit.edu
- Isabelle Chamberlain (AEE) - ichamberlain2023@my.fit.edu
- Joshua Huser (AEE) - jhuser2023@my.fit.edu
- Will Emerson (AEE) - wemerson2023@my.fit.edu
- Tharushi Pelendagama (AEE) - tpelendagama2023@my.fit.edu

3. Faculty Advisor

Firat Irmak, PhD - firmak@fit.edu

4. Client

Firat Irmak, PhD - Aerospace Engineering Department, Florida Institute of Technology

5. Meeting Date(s) with Advisor/Client for Developing this Plan

- Monday, August 31, 2026
- Monday, September 14, 2026
- Monday, September 28, 2026
- Monday, October 12, 2026
- Monday, October 26, 2026
- Monday, November 9, 2026
- Monday, November 23, 2026

6. Goal and Motivation

Team S.H.I.E.L.D. is challenged to design, build, and test a counter-UAS system for small, low-altitude aerial targets. This system shall be capable of autonomously detecting, tracking, and intercepting an aerial target using onboard sensing and computing.

7. Approach (Key Features)

- Keep a designated area under constant watch, and count on SHIELD to respond the moment an unauthorized drone comes within range. Once a target is spotted, SHIELD locks on, follows its every move, and intercepts it within 20 meters. No manual aiming or tracking required.

- Decide exactly how much control you want. Engage manual override to personally direct the response for peace of mind, or let SHIELD's onboard AI handle detection and engagement entirely on its own. Stay hands-on or hands-off depending on the situation.
- Once SHIELD locks onto a target, it stays locked on, tracking through brief obstructions or awkward angles instead of losing the target and having to reacquire it from scratch.

8. Novel Features/Functionalities

The semi-autonomous detection and engagement is a novel functionality of SHIELD. SHIELD automatically detects and prompts an operator to engage. Once the human approves the engagement, the drone can track and intercept a target using onboard edge computing, rather than relying on a ground operator for control. Because of this, SHIELD is more difficult to jam or hack once it is pursuing a target.

9. Algorithms and Tools

- OpenCV: video frame preprocessing and per-track Kalman filtering (smoothing + motion prediction)
- Ultralytics YOLO (yolo11n.pt): real-time object detection
- ByteTrack (bundled with Ultralytics): persistent multi-object track IDs
- Unity (Editor 6000.3.11f1): "SHIELD Virtual Camera" stand-in camera feed for PC-based emulation, streamed to the Python pipeline over TCP
- Raspberry Pi 5 + AI HAT+ (Hailo accelerator) + Camera Module 3: target hardware platform
- STE (SHIELD Test Environment): in-house VB.NET/C# test harness for running scripted test cases (Test Scripts/FMS, Test Scripts/SYS)

10. Technical Challenges

- Running detection and tracking efficiently on the Raspberry Pi 5 + AI HAT+ (Hailo accelerator) edge hardware: the pipeline is currently only validated CPU-only on a Windows PC, and moving to the power- and thermal-constrained onboard hardware while keeping frame rate high enough to track a moving target in real time is still an open problem.
- Making that edge inference reliable enough to trust for semi-autonomous operation: since the drone can engage without a human in the loop (other than to abort), detection and tracking need to hold up consistently on the embedded hardware and in real-world conditions, not just in ideal PC-based emulation, before the system can be trusted to run on its own.
- Training a custom drone/balloon detection model to replace the current stock COCO-pretrained model, so the ≥90% confidence threshold (SYS.07) is met on the targets, which is a high threshold that may have to be revisited later, especially because of the transition from object detection in a virtual environment to a real-life environment.
- Integrating all the code that runs on emulation onto the drone prototype once it is created will be a challenge, as will accurately simulating the drone's physics in Unity for testing.

11. Milestone 1 (Sep 28) - Itemized Tasks

- Compare and select technical tools for detection, tracking, and (once defined) interception subsystems.
- Provide "hello world" demos evaluating candidate tools - already partially done: the repo's emulation pipeline runs YOLO detection + ByteTrack + Kalman smoothing against a webcam or the Unity virtual camera.
- Resolve initial technical challenges identified above.
- Compare and select collaboration tools for software development, documents/presentations, communication, and task scheduling.
- Create Requirement Document.
- Create Design Document.
- Create Test Plan.

12. Milestone 2 (Oct 26) - Itemized Tasks

- Complete development on STE (SHIELD Test Environment) program.
- Find, combine, and train a custom drone/balloon detection model.
- Add moving target GameObject instantiation to testlib and motion scripting to the Unity test scenes.
- Write and implement STE test scripts to verify system requirements and initial requirements in requirement document.
- Design and begin implementing the manual override control interface.

13. Milestone 3 (Nov 23) - Itemized Tasks

- Integrate detection/tracking output with the interception subsystem (emulation only) once it exists.
- Implement "Hardware" platform mode (Raspberry Pi 5 + AI HAT+) and test on drone prototype, if built.
- Run the custom-trained model against the full set of Unity lighting/environment scenes and evaluate confidence against the SYS.07 ≥90% threshold.
- Have all existing requirements traced to STE test scripts.
- Conduct full end-to-end system test and demo on emulation.

14. Task Matrix for Milestone 1

Task	Bryson Noble (SWE)
Compare and select technical tools	Analyze efficiency of YOLOv11n to detect a drone against other versions/sizes
"Hello world" demos	Completed 08/26 using the Unity virtual camera and a preliminary model and two flying objects in the scene.
Resolve technical challenges	Improve model to have higher confidence

Task	**Bryson Noble (SWE)**
Compare and select collaboration tools	Using Visual Studio for development, GitHub for version control, Microsoft Teams for collaboration with AEE team.
Requirement Document	Begin subsystem requirements
Design Document	Create document
Test Plan	Create document + STE

15. Approval from Faculty Advisor

"I have discussed with the team and approve this project plan. I will evaluate the progress and assign a grade for each of the three milestones."

Signature: ________________________________ Date: ________