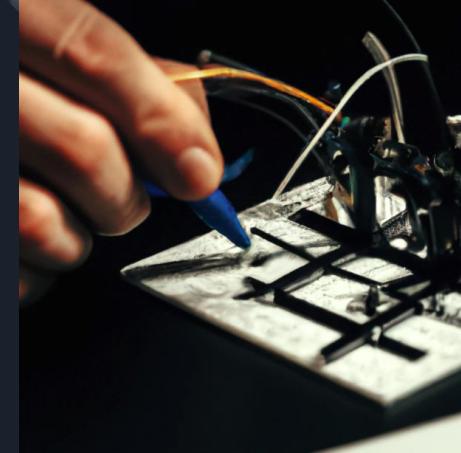




Hand Held GUI for Drones



Team Mango - Ayan, Harinie, Ronak, Sriya



Problem Statement

Problem(s):

- Military operations require reliable & efficient means of conducting operations in hostile environments. Traditional methods can be risky and costly.
- Using drones is becoming an increasingly attractive alternative. However, operating them requires specialized skills which may not be available or practical.
- Currently, disconnect exists between information from drone's sensors and user's actions.

Aim:

- App-based solution that provides a seamless interface between the drone and the user.
- The handheld GUI will offer various modes & features to allow the drone specialized functions with user-friendly interface.



Motivation

- Allows users with little or **no prior drone experience** to operate drones in high-pace and quick-thinking situations.
- More **user-friendly** and **accessible** method of controlling and navigating drone.
- Can be used from **remote** locations for **rapid** responses.



Application - Real Life Use Case

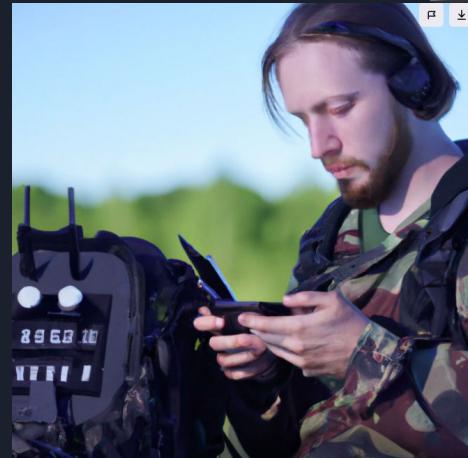
- Attack (Anti-Drone)
- Search
- Intelligence Gathering
- Surveillance





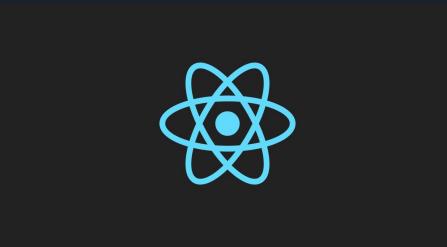
Target Users

- Inexperienced personnel
- Military personnel (soldiers, etc)





Technical Architecture



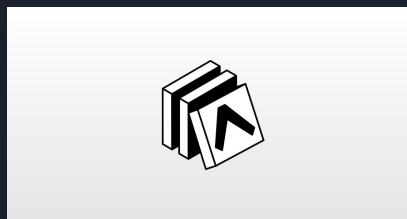
React Native



JavaScript



TypeScript



Expo

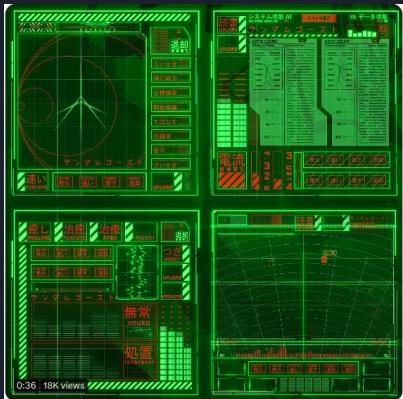


Taskade



Gitlab

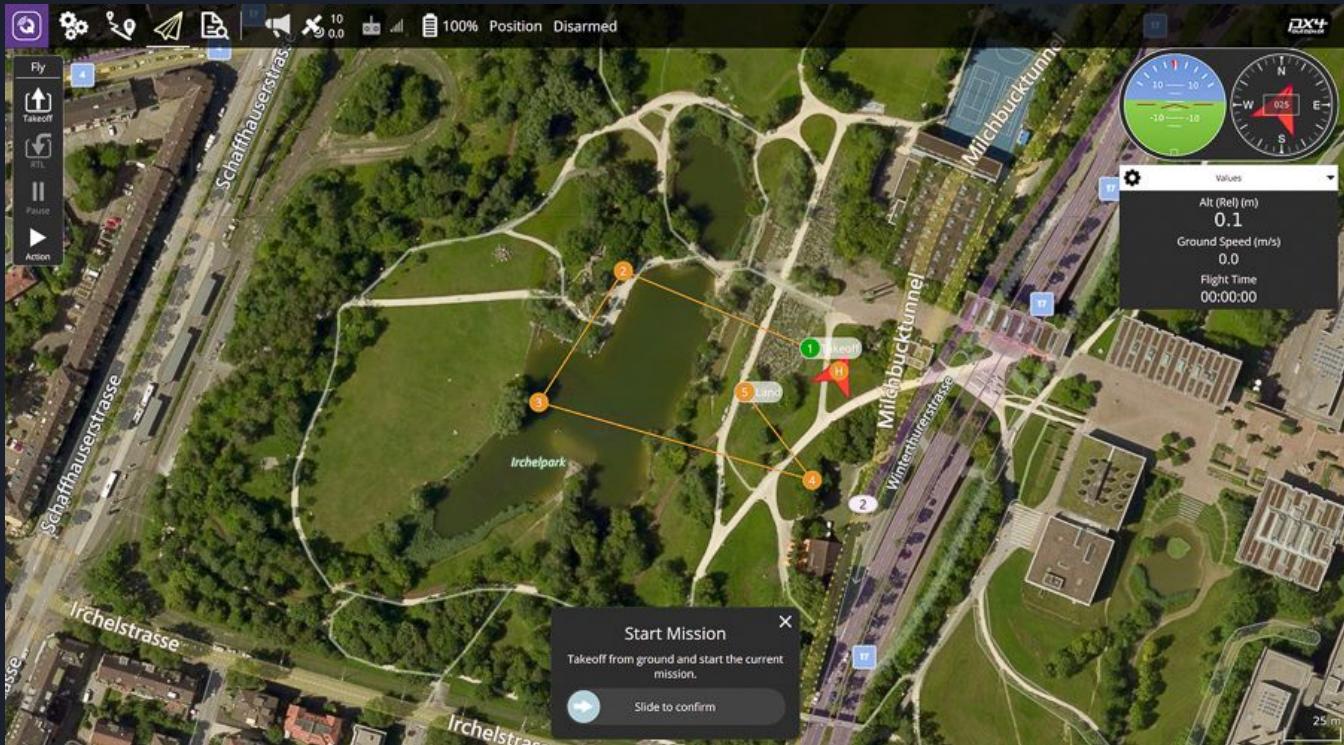
Research/Study



Research/Study - ArduPilot Mission Planner



Research/Study - QGroundControl



Scoping - The Development Timeline

- Sprint 0 - Project finalization and motivation (Jan 19)
- Sprint 1 - Discussing scope and non-scope (Jan 24)
- Sprint 2 - Features and modes brainstorm (Feb 1)
- Sprint 2 - Describing GUI screen requirements (Feb 7)
- Sprint 3 - Developed basic app with two screens (Feb 13)
- Sprint 3 - Design of flowcharts to visualize user flow (Feb 20)
- Sprint 3 and 4 - Development of Phase 1 of the app (Feb 27)
- Sprint 4 - Focus on Phase 1 and 2, prepare for R1 (Mar 7)
- Sprint 5 - Feedback of R1, Adding phases (2-6), re-scoping, fine tuning of map functionality, drone selection on map (Mar 24)
- Sprint 6 - UI Aesthetics, settings window, alert bar (Apr 11)
- Sprint 7 - Backend, simulation, documentation, presentation (Apr 18)



Completed Tasks

Sprint 0

- Discussed several project ideas
- Finalized on the hand-held GUI for drones



Sprint 1

- Developing project outline
- Identifying intended users and applications
- Scope and non-scope discussion



Completed Tasks

Sprint 2

- Developed features document
- Feature selection and modes

Sprint 3

- Framework review - Suitable tech stacks
- Designed flowchart and screens for phases
- Developed phase 1



React Native



JavaScript



TypeScript



Expo



Taskade



Gitlab

Completed Tasks

Sprint 4

- Development of remaining phases (Follows from phase 1)

Progress since R1

- Added compass, magnetometer to map
- Added drone selection on the map end and switching between targets

Sprint 5

- Made signals for ground and drone
- Fixed map navigation issues - zoom
- Made UI changes based on feedback from client



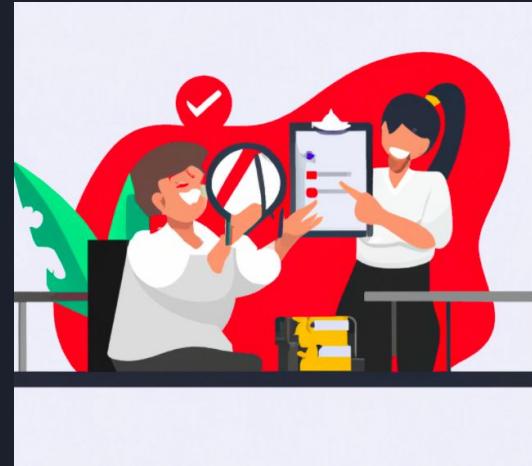
Completed Tasks

Sprint 6

- Made swipe bars for drone actions
- Made the UI more intuitive based on client feedback
- Started working on the simulation of drone encounter

Sprint 7

- Implemented last few changes needed by client
- Revamped the aesthetics for a better presentation
- Completed the simulation for the demonstration
- Finished the documentation



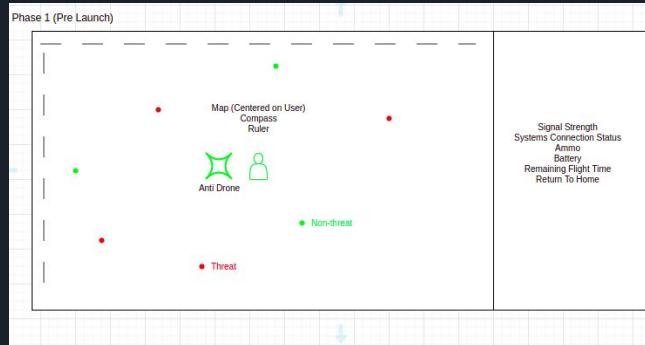


Main Features in our GUI

- Multiple Screens
- Status Bar showcasing:
 - Signal Strength
 - Battery
 - Ammunition
 - Launch/ Return to Home
 - System Communication Status (Radar, Drone, etc)
- Map showcasing:
 - Color coded objects using Threat Matrix
 - Markers for Threats,, Non-Threats, Anti-Drone
- Camera (activated when within 60 metres of Object (Threat))
- Sliding buttons for follow, kill, abort, navigate to kill.
- Settings screen and RTH button
- Alert bar on the top of screen to show important information

Different Screens/Scenarios :

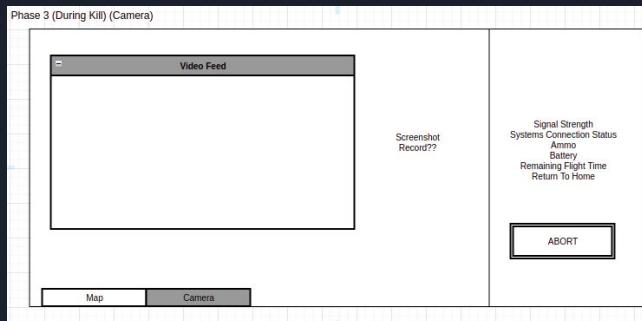
01



02



03



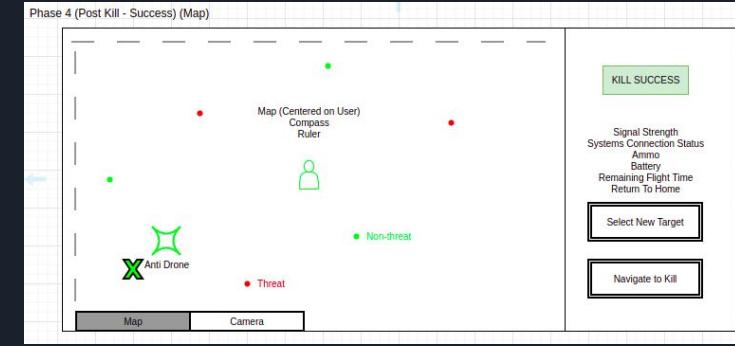
04



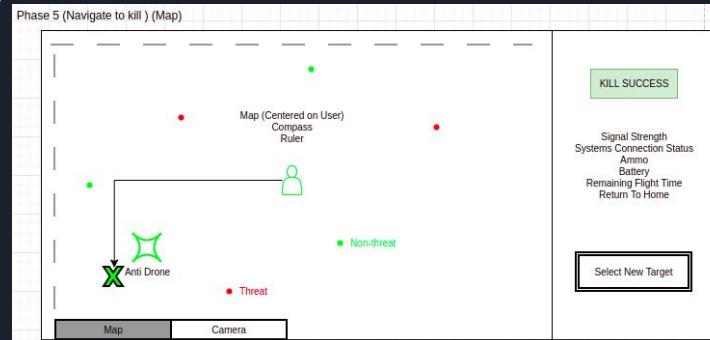
05



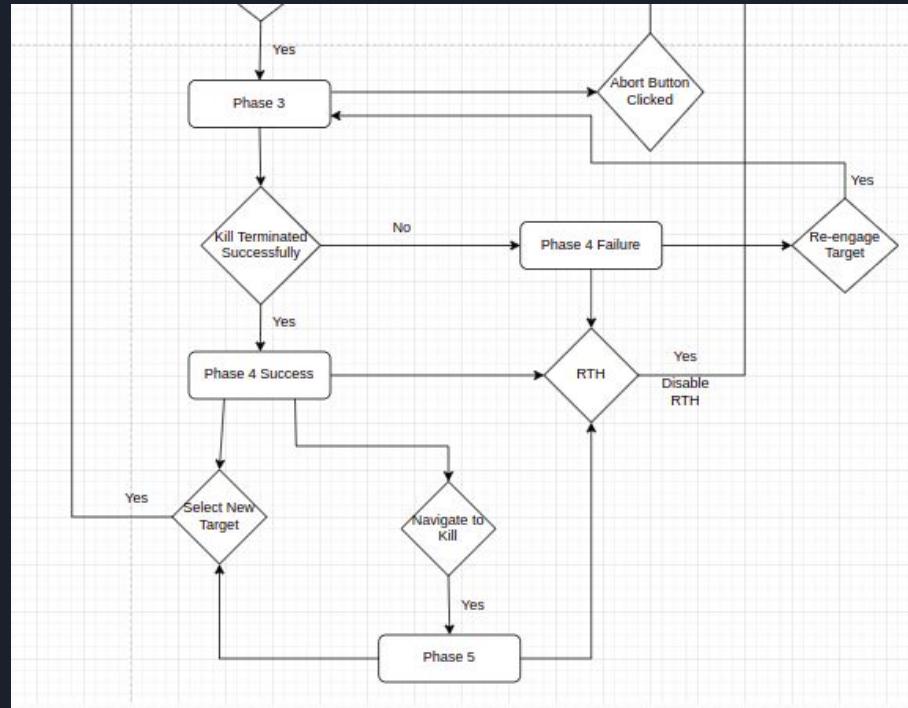
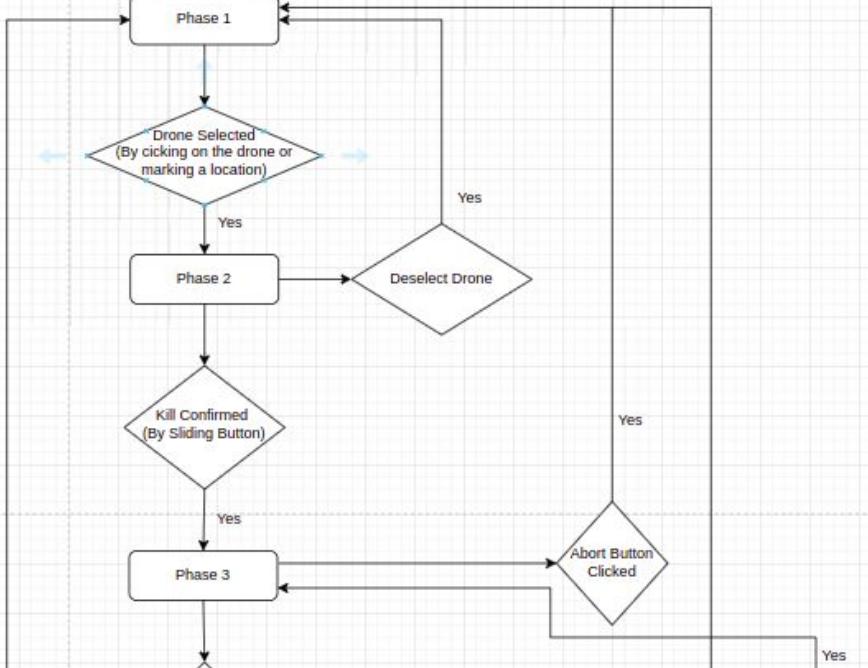
06

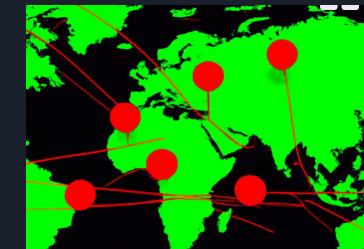
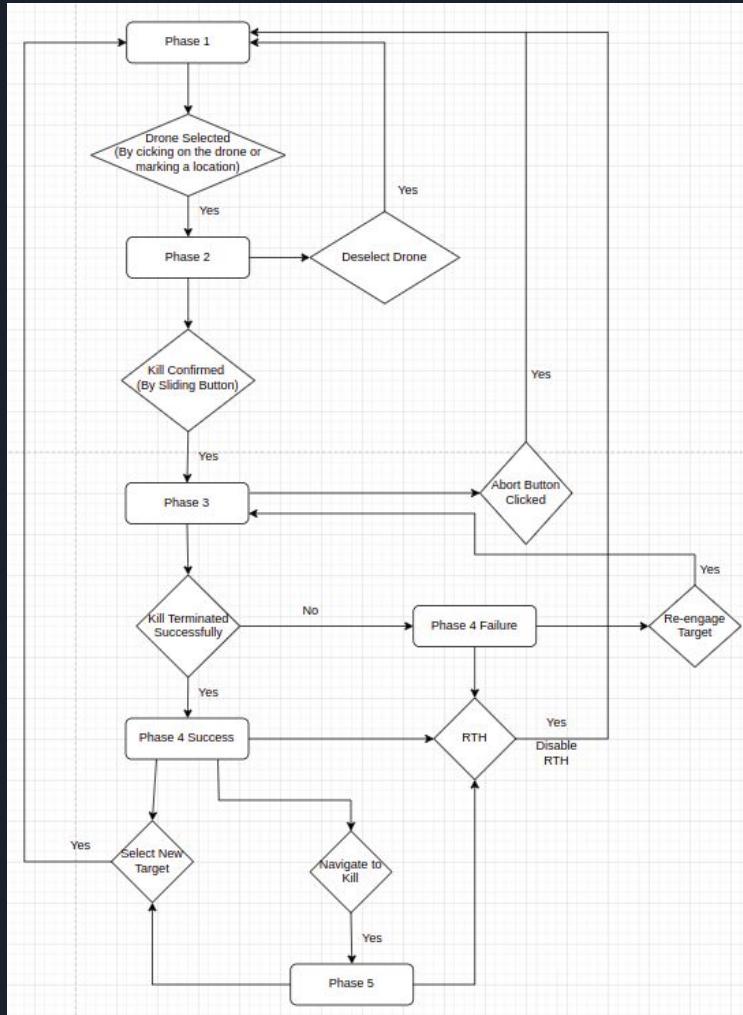


07



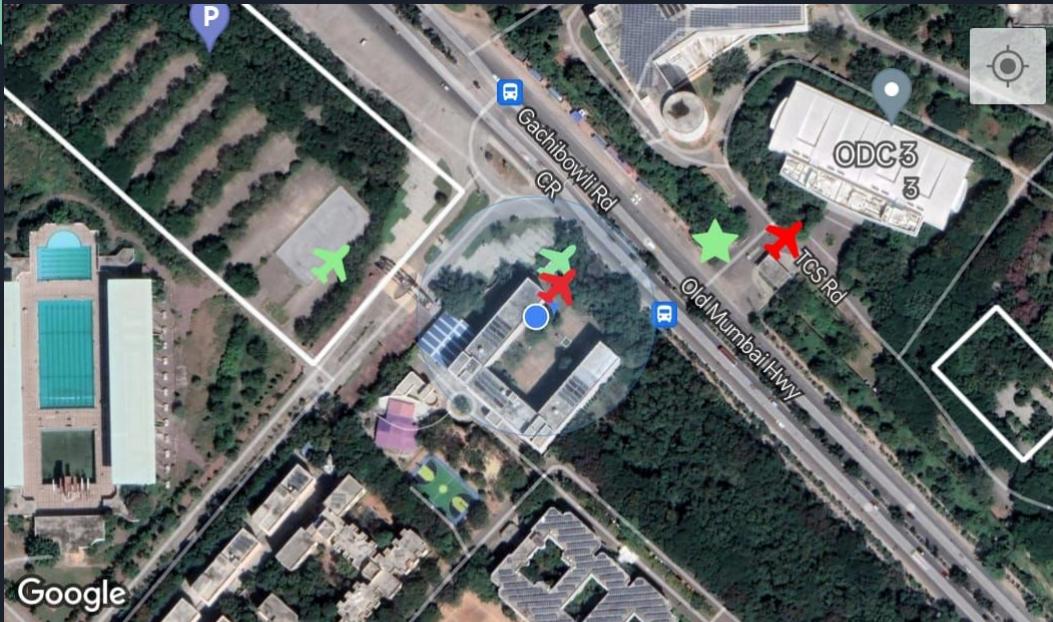
Flowchart:





Detailed Progress since R1

Then



Map



Camera

Status

Signal Strength:

Ammo:

Battery:

Flight Time Left : 20:00

Return to Home



Phase 1

Drone ready for Take-Off

The map displays a residential area with a large stadium and a road network. Several icons are overlaid on the map, including a red airplane icon with a black outline, a green house icon, a yellow square icon with a black outline, a green person icon, a red star icon, and a red airplane icon with a black outline. A blue diamond icon and a green location pin are also visible. The word "Google" is in the bottom left corner.

Status

- Signal strength icon
- Flight mode icon
- Drone battery icon
- Drone orientation icon
- Flight time: 20:00

Map Camera

Home



Phase 2

Distance to Target: 60 m

Status

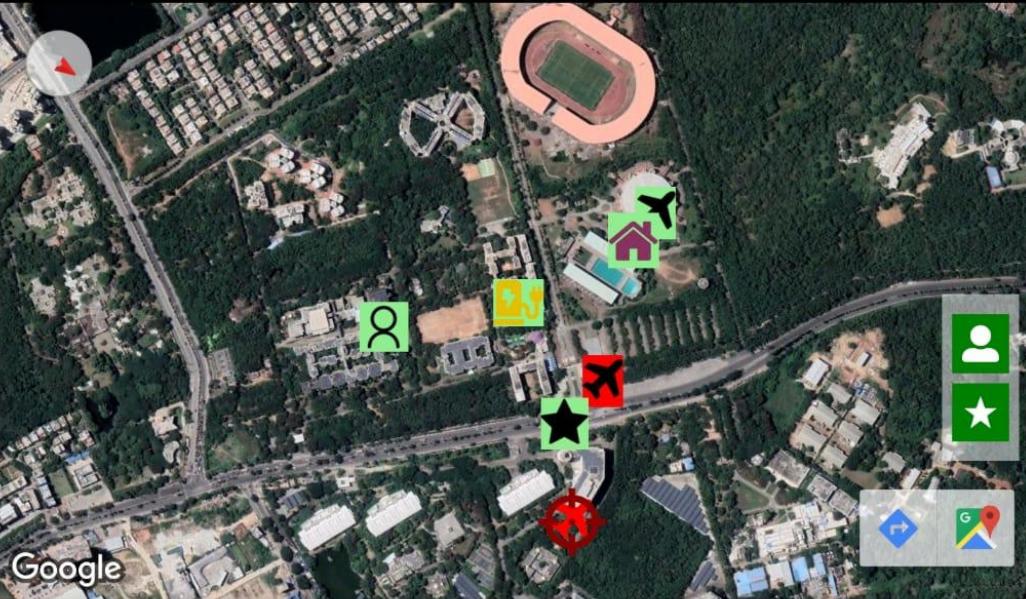
: 20:00

Google

Map Camera

Phase 3

Distance to Target: 40 m



Status

- Signal Strength: Bar icon
- Network: Wi-Fi icon
- Location: GPS icon
- Time: 20:00

Red button Swipe to Kill

Green button Swipe to Abort

Map Camera Home

Phase 4

Distance to Target: 20 m

Status

Swipe to Follow

Swipe to Abort

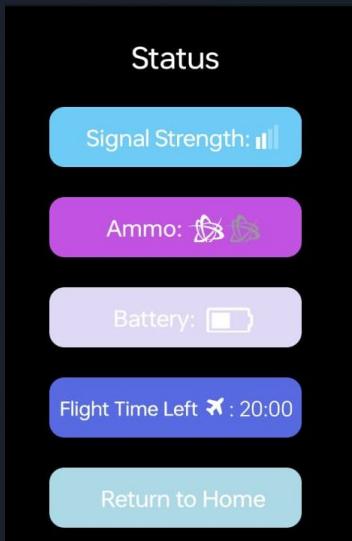
Google

Map Camera

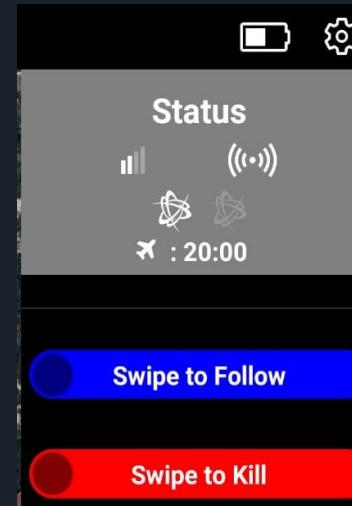
Detailed Progress since R1

- Separated icons and signals from drone actions (buttons)
- Developed swipe buttons for kill, follow and abort

Then

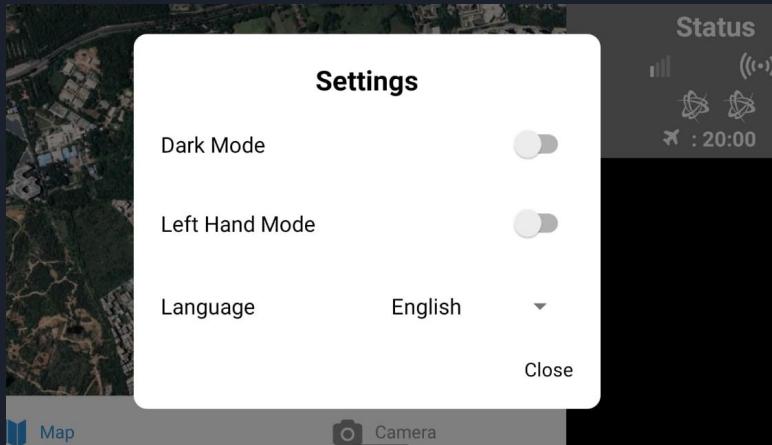


Now



Detailed Progress since R1

- Settings Tab



- Alert Bar

- Phase 1: ready for takeoff
- Phase 2, 3, 4: distance to target
- Phase 5: Success/ Failure
- Phase 6: Recovery Priority
- Also displays critical messages such as 'Low battery level'



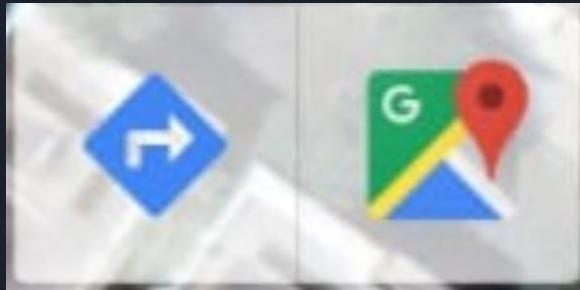
Drone ready for Take-Off

SUCCESS



Detailed Progress since R1

- Implemented magnetometer for map
- Added compass and directions buttons





Detailed Progress since R1

- Enabled select drone functionality to choose the target and drone switching by tapping
- Added cross-hair to the selected drone target

Then



Now



Detailed Progress since R1

- Added icons for home base and charging station on the map
- Allowed capability to center map on enemy drone or anti-drone



Then



Now



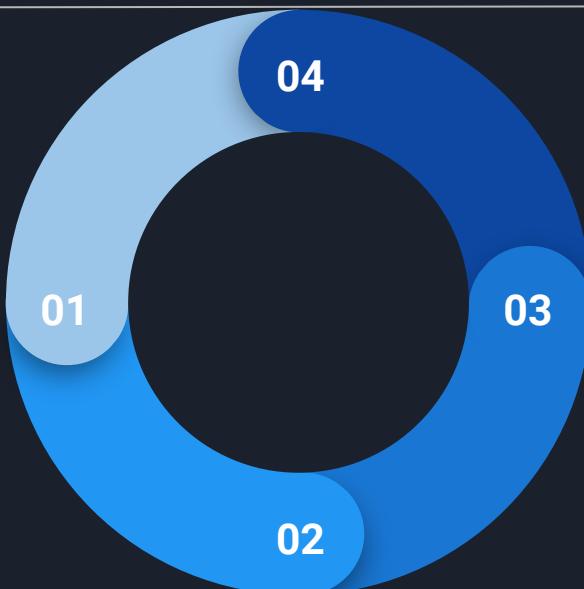


Detailed Progress since R1

- Simulation: The movement of drones and the antidrone is simulated using a python script which creates a socketio server and publishes antidrone and ground control updates.
- The simulation was integrated into the app in a way that the app does not know whether the values its getting are mocked or not. This means that a drop in replacement for interfacing with actual drones and antidrones can be made and integrated by making no changes to the app.

Future Scope

Sensor Access,
Read Input from USB



Estimation of RTH

Refine, Dev + Test
Sprint

Drone effectiveness
observation



Challenges

- Our first and biggest challenge was to ensure that the data shown on the GUI and its working was the most optimal and the most necessary for the intended users (soldiers). To achieve this the GUI was designed considering all possible situations that could occur and a phase wise approach was used.
- A very crucial aspect of this project was to make an interface that would be easy to use and intuitive in its functioning. To that end, every UI decision, the sizes of buttons, the zoom level of the map, the ease of performing actions like engage and abort were something that we had to continuously work on improving. There was not a meeting that went by where we didn't change at least some aspect of the aesthetic or UX.
- Working on multiple components in the frontend simultaneously by different team members poses a challenge when integrating and merging the code together. It requires extra effort to merge even with communication between the team.



THANK
YOOU!

