Autonomous Cognitive Agent Flying Aircraft in X-Plane

Milestone 2

Avery Mackey - amackey2023@my.fit.edu
Dylan Ostolaza - dostolaza2024@my.fit.edu
Zachary Reese - zreese2023@my.fit.edu
Brianna Marshall - bmarshall2022@my.fit.edu

Milestone 2 Progress Matrix

Task: *Implement, test, and demo…*	**Completion**	**To-Do**	**Avery Mackey**	**Dylan Ostolaza**	**Zachary Reese**	**Brianna Marshall**
Simulate and communicate equipment failures to SOAR	100%	None	0%	0%	100%	0%
Follow a flight plan with no complications	85%	Handle waypoints, GUI updates	0%	0%	20%	80%
Develop SOAR rules for failure recognition and response	100%	None	95%	0%	5%	0%
Execute and verify aircraft failure responses	100%	None	0%	100%	0%	0%

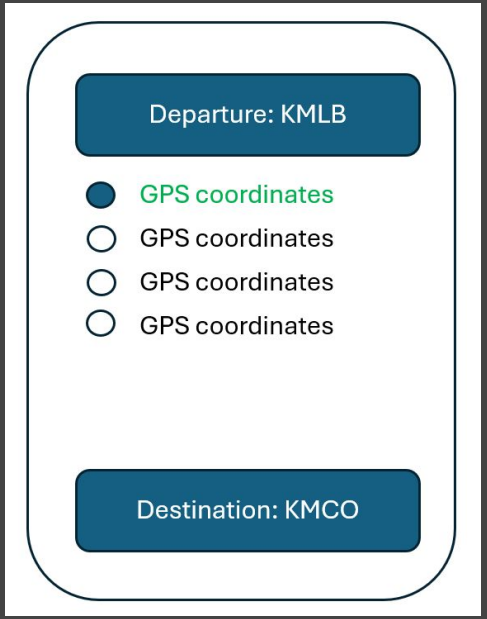
Autonomous Flight Navigation

- Implemented autonomous flight from takeoff to landing
- Added SOAR logic for cruising, descent, and vertical landing
- Integrated new sensor inputs for flight navigation and control
- Automated autopilot controls and heading adjustments
- Improved communication reliability between Java and X-Plane
- Remaining:
  - Implement waypoint navigation and update the GUI to display flight plan information

Challenges

- Multiple threads sharing XPlaneConnect caused overlapping responses
  - Implemented synchronized requests to prevent communication conflicts

Proposed Flight Planning GUI



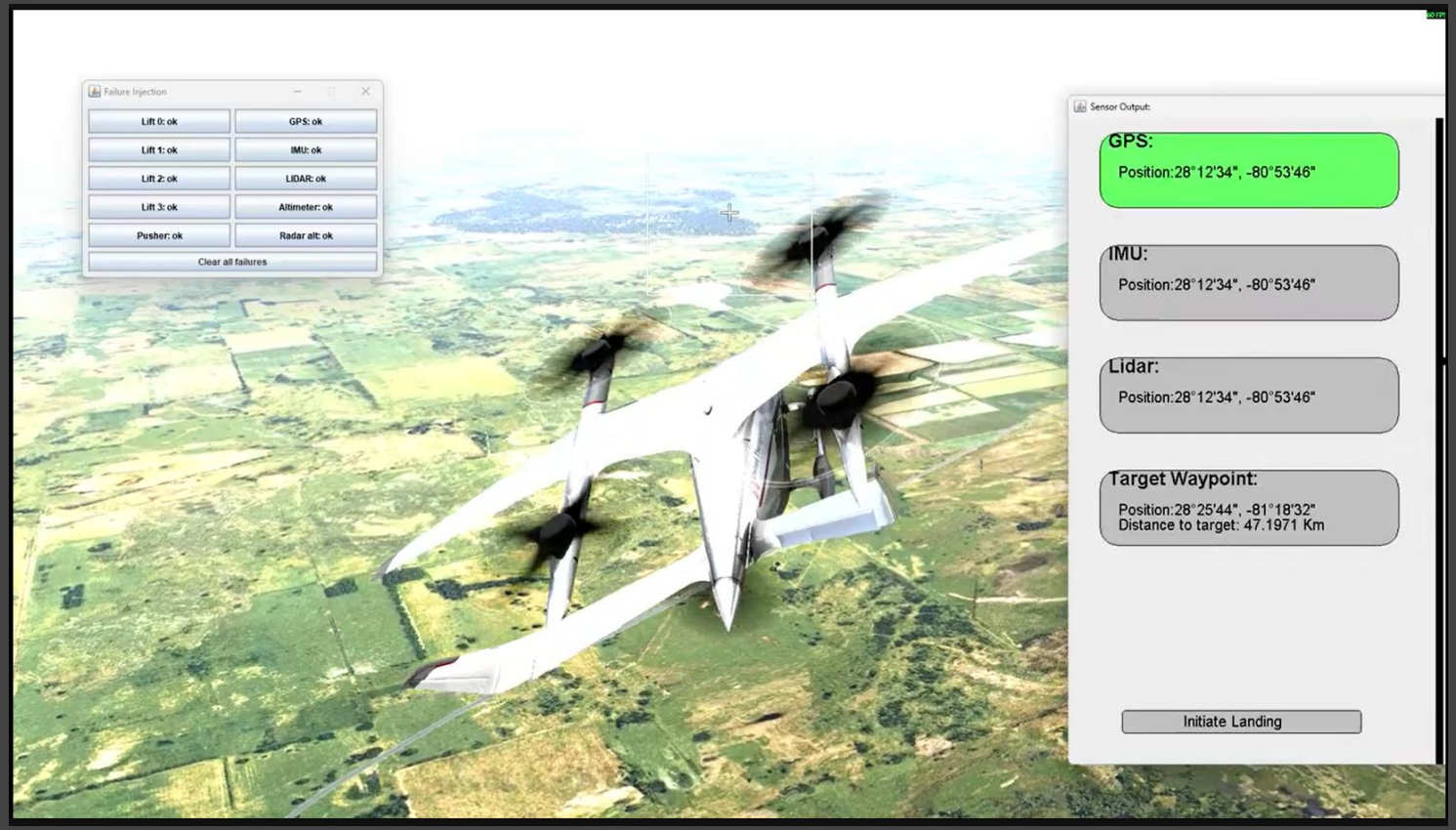
Autonomous Flight Navigation - Demo

Equipment Failure Simulation

- Created a new GUI window for simulating aircraft equipment failures
- Implemented FailureInjector.java to communicate failures to SOAR
- Added failure controls for lift rotors, pusher motor, and navigation sensors
- Supported four sensor failure modes: bias, drift, freeze, and dropout
- Communicated active failures through the SOAR input link

Challenges

- Needed a consistent method to communicate different failures to SOAR
  - Organized all equipment failures under a common SOAR input-link structure



Equipment Failure Simulation - Demo

SOAR Failure Recognition & Response

- Implemented failure-handling.soar to recognize equipment and sensor failures
- Evaluates the aircraft's remaining capabilities after a failure
- Determines an appropriate response based on failure type and flight phase
- Switches to backup sensors when primary navigation sensors fail
- Initiates precautionary or emergency landings when necessary
- Establishes operator priorities to prevent conflicts between emergency responses and normal flight operations

Challenges

- Simultaneous SOAR operators caused decision conflicts
  - Established operator priorities to ensure emergency responses take precedence
- Emergency landings followed normal destination-based airspeed logic
  - Added an override to immediately reduce target airspeed for emergency landings

Failure Response and Verification

- Implemented FailureResponseHandler.java to process SOAR's selected responses
- Processes SOAR's selected responses and verifies command execution in X-Plane
- Verifies aircraft responses by reading X-Plane flight data
- Checks throttle settings, autopilot status, and active navigation sensors
- Records verification results as PASS/FAIL in a CSV file
- Performs verification on a separate thread to avoid interrupting the agent

Challenges

- Issuing a command did not guarantee the aircraft executed it
  - Added checks to compare the expected response against actual aircraft data
- Some responses cannot be directly observed in X-Plane
  - Records these responses as informational results

Next Milestone Task Matrix

Task: *Implement, test, and demo…*	**Avery Mackey**	**Dylan Ostolaza**	**Zachary Reese**	**Brianna Marshall**
Automate flight planning with waypoints	10%	10%	10%	70%
Agent handles multiple simultaneous equipment failures	70%	10%	10%	10%
Execute complete flight plans with equipment failures	10%	70%	10%	10%
M2 improvements: Altitude control, landing, GUI updates, and FAA standards verification	10%	10%	70%	10%

Any Questions?