Autonomous Cognitive Agent Flying Aircraft in X-Plane

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

Advisor / Client

Dr. Siddhartha Bhattacharyya

- Department of Electrical Engineering and Computer Science
- Program Chair Computer Science and Software Engineering

Meetings

- August 27 @ 10 a.m.
- August 28 @ 2 p.m.

Goals & Motivation

Goals

- Extend the existing X-Plane and SOAR framework
- Develop a cognitive agent that can adapt the flight plan in response to aircraft equipment failures

Motivation

- Create a safe and adaptable autonomous flight system
- Current models are adapted to specific problems with static flight plans

Key Features

The user can...

- Define the original departure and destination airports for the cognitive agent's flight route
- Create and simulate an aircraft equipment failure
- Create weather and human-population density scenarios
- Observe the flight route change in response to new conditions

Novel Features

- **Equipment failure handling**
  - Extends the system, beyond normal flight operations by allowing it to operate under degraded aircraft conditions
- **Dynamic flight routes**
  - The planning flight route can change with new conditions

Algorithms & Tools

- **SOAR Production Rules / SOAR Architecture**
  - Agent response and memory updates
- **IntelliJ**
  - Java integrated development environment
- **Java**
  - Primary programming language
- **X-Plane Datarefs**
  - X-Plane flight data access and control
- **Maven**
  - Java project configuration and dependency-management tool
- **XPC Plugin (X-Plane Control)**
  - Software control and communication with X-Plane

Technical Challenges

- **SOAR architecture**
  - New cognitive architecture
  - Learn production rule system, memory model, and development workflow
- **X-Plane & Aircraft Behavior**
  - Aircraft systems, controls, physics, and engineering behavior
  - Model failures and interpret aircraft's response
- **Existing Codebase**
  - Understand current implementation and communication among SOAR, Java, and X-Plane
  - Extend the code without breaking the existing framework

1. **Compare & Select Technical Tools for SOAR, X-Plane, and Secant**
    - Provide (“Hello World”) demos to evaluate chosen tools
2. **Resolve initial technical challenges**
    - Soar architecture
    - X-Plane and aircraft behavior
    - Existing codebase
3. **Compare and select collaboration tools**
    - GitHub
    - Google Drive
    - Group chat & Email
    - Jira
4. **Create Requirement Document**
5. **Create Design Document**
6. **Create Test Plan**

Milestone 1

1. ***Implement, test, and demo*** **-** Triggering and communicating equipment failure from X-Plane to the SOAR agent

2. ***Implement, test, and demo*** **-** Agent following a complete flight plan in X-Plane with NO complications

3. ***Implement, test, and demo*** **-** Agent recognizes and responds to equipment failure

4. ***Implement, test, and demo*** **-** Agent issues appropriate commands to the aircraft

Milestone 2

1. ***Implement, test, and demo*** **-** Failure generation system with multiple component failures

2. ***Implement, test, and demo*** **-** Multiple failures and solution visualization on the GUI

3. ***Implement, test, and demo*** **-** Cognitive agent response to multiple component failures in parallel

4. ***Implement, test, and demo*** **-** Complete flight plan with one or more equipment failures

Milestone 3

Task Matrix

Task	Dylan	Zach	Avery	Brianna
Compare and Select Technical Tools	25%	25%	25%	25%
“Hello World” Demos	30%	30%	20%	20%
Resolve Technical Challenges	25%	25%	25%	25%
Compare & Select Collaboration Tools	Documents / Presentations	Software	Communication	Task Calendar
Requirement Document	20%	20%	20%	40%
Design Document	20%	40%	20%	20%
Test Plan	40%	20%	20%	20%

Any Questions?