

# Proposed enhancement to FlightGear.

By: Group 3 - Project Flight Forge

Video URL: <https://youtu.be/wxbWNJUcNms>



# Roles and responsibilities

- ▶ Leader: Alan Teng - Introduction
- ▶ Presenter: Divaydeep Singh -> SAAM Analysis.
- ▶ Presenter: Lucas Coster - Motivation, Implementation, Affects, Risks, Conclusion
- ▶ Team member: Donovan Bisson - Implementation #2, Testing
- ▶ Team member: Antonio Sousa-Dias -> Use Cases.
- ▶ Team member: Zhonqi Xu - Implementation #1



# Our Enhancement

## Customization Module

### Motivation

- User Experience

### Interactions

- GUI
- Cockpit
- Aircraft
- More



# Current State



**10 Subsystems**



**Relevant Information**

Cockpit  
GUI



# Implementation Proposal

Repository  
Style

DataBase

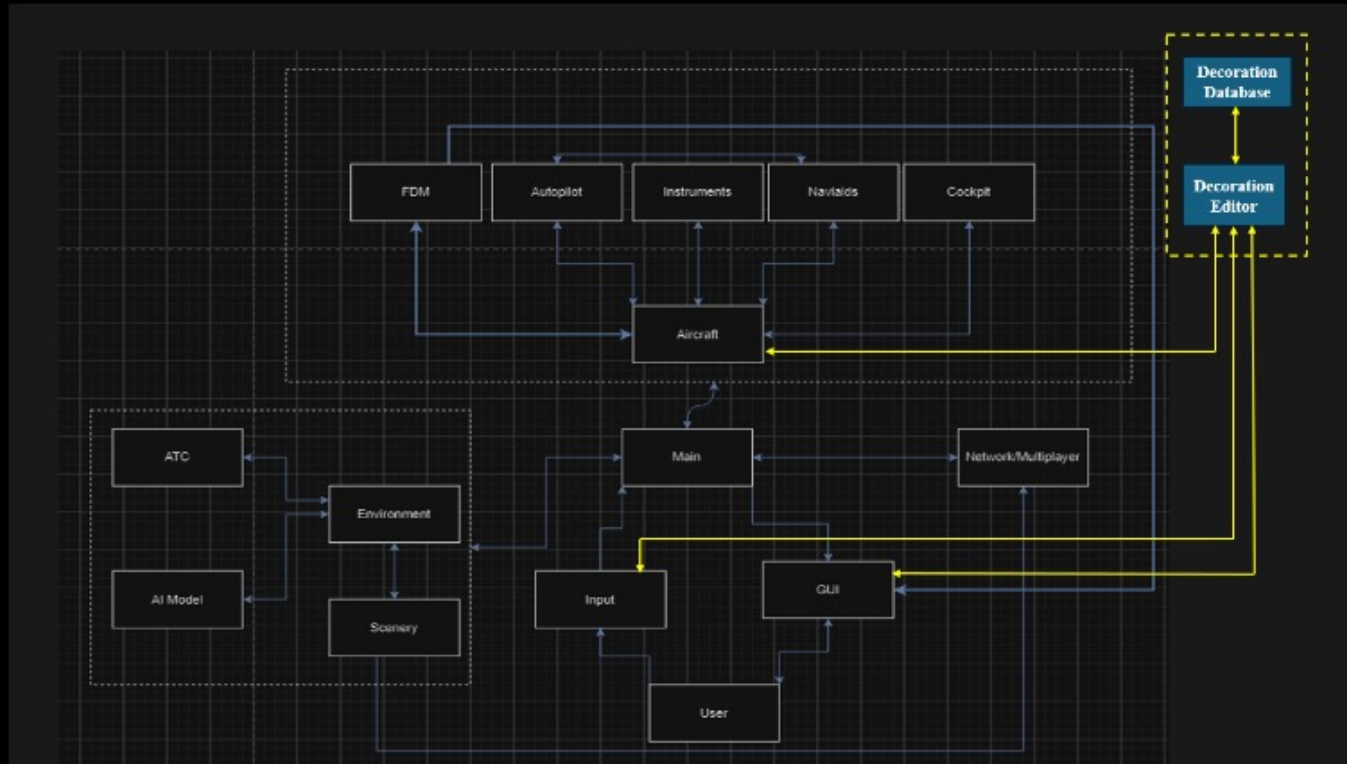




# Implementation #1: In-Game editor

- ▶ Overview
- ▶ Subsystems
  - ▶ Decoration Editor
  - ▶ Decoration
- ▶ Dependencies
  - ▶ Aircraft
  - ▶ GUI
  - ▶ Input
  - ▶ Decoration -> Decoration Editor





# Implementation #1: Diagram



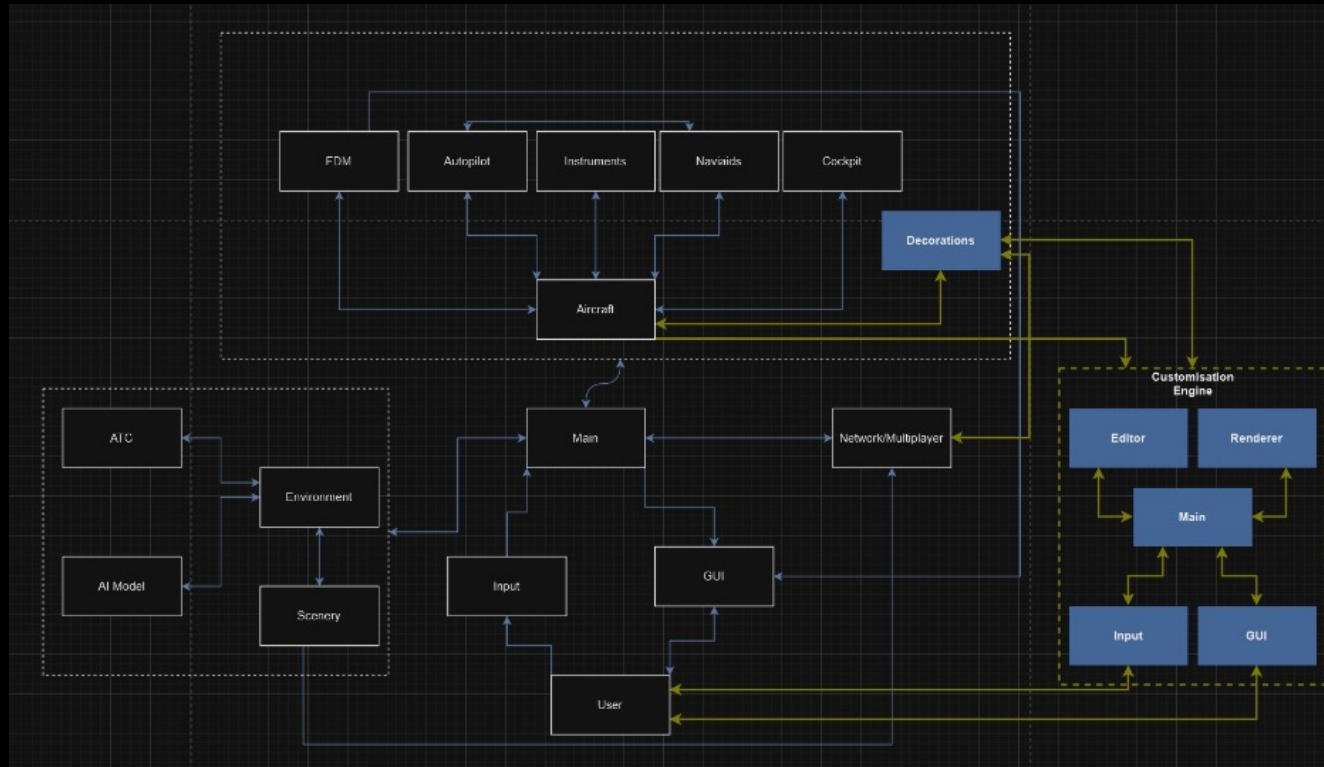


# Implementation #2: New Module

- ▶ Decoration Editor Module
- ▶ Subsystems
  - ▶ Decorations
  - ▶ Editor
  - ▶ Renderer
- ▶ Dependencies
  - ▶ Aircraft
  - ▶ User
  - ▶ Multiplayer







# Implementation #2: Diagram



# SAAM Analysis

- ▶ Two approaches for implementing plane customization and cockpit building.
- ▶ Modifying existing source code to implement an in-game editor or implementing the enhancement through a new module.
- ▶ Two stakeholders identified, users and developers.
- ▶ Users have usability, performance, reliability, and compatibility as the most important non-functional requirements.
- ▶ Developers have performance, interoperability, security, and maintainability as most important non-functional requirements.



# Stakeholders: Users



## Usability:

Implementation #1 provides easier learning to users as it operates within the system.



## Performance:

Implementation #2 optimizes performance by adding slight computational overhead.  
Implementation #1 reduces performance as code is added within main loop.



## Reliability:

Implementation #2 maintains reliability as issues within the external module do not impact overall system.



## Compatibility:

Implementation #2 maintains compatibility with other FlightGear versions as users can download the external module if they want.



# Stakeholders: Developers



## Performance

Implementation #2 maintains performance.



## Interoperability

Option #2 can be easily made compatible with other modules and files.



## Security

Option #1 operates within FlightGear's environment. No new security concerns.

Option #2 needs its own security considerations.



## Maintainability

Option #2 is easier to maintain and update as a standalone module.



# Effects on Quality Attributes



**Maintainability:** Proposed enhancement would increase the complexity of the system. Maintenance tasks could potentially become more challenging.



**Evolvability:** Increased evolvability by providing a flexible framework for adding new customization features in the future.



**Testability:** New module would necessitate the development of comprehensive testing procedures to ensure stability and functionality.



**Performance:** Rendering and simulation speed could be negatively impacted.





# Potential Risks

- ▶ Security
- ▶ Maintainability
- ▶ Performance
- ▶ User Experience



# Testing



DECORATION  
INFO



NEW TESTS



WITHIN-SIM  
EDITOR



EXTERNAL  
EDITOR

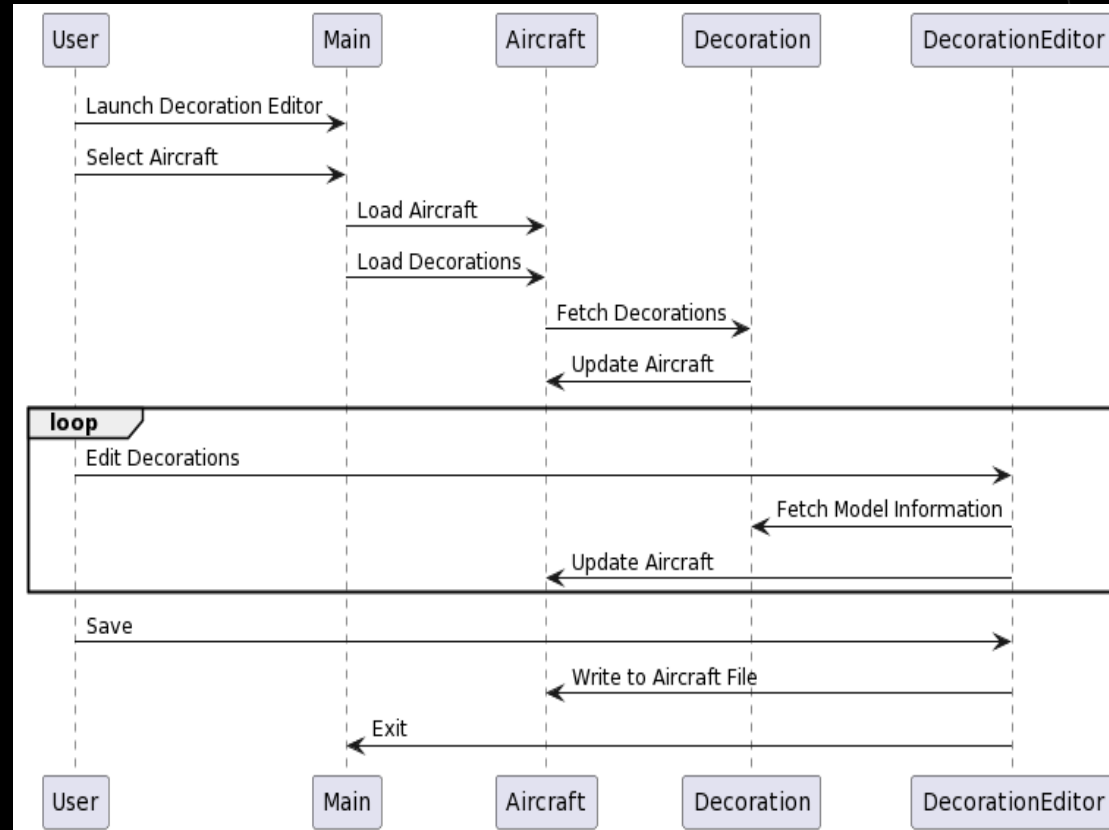


IMPACT OF  
CHANGES



# Use case

## ► Editing decorations





# Concurrency and Team Issues

- ▶ Lack of Code Consistency
- ▶ Multiple Different Programmers
- ▶ Coding Styles and Comments



# Conclusions and lessons learned



**Proposed Enhancement**



**Lessons Learned**

User Experience  
Stakeholders  
Feasibility

