Plugin-Based Drone Show System

Overview

This project implements a modular architecture for validating and generating drone shows using multiple independent plugins. Each plugin focuses on a different aspect, including:

- Drone program validation
- DSL-based figure validation
- Show-level DSL generation
- Proposal validation

Plugins Overview

Drone Program Plugins

`DroneScript(Language_1)` (droneone-plugin)

- Validates drone program scripts.
- No code generation.
- UI: `DroneValidator_L1_UI`
- Class: `DroneValidator`

`Drone_Script(Language_2)` (dronetwo-plugin)

- Validates drone program scripts.
- Code generation supported.
- UI: `DroneValidator_L2_UI`
- Classes:
  - `DroneValidator`
  - `DroneLanguagePlugin`
  - `generateCode(...)` per model

DSL Figure Description Plugin (`DSL-PLUGIN-1.1.23`)

- Validates figure descriptions written in a custom DSL.
- Based on ANTLR grammars: `DSL_FIGURE.g4`, `DSL_GRAMMAR.g4`
- Classes:
  - `FigureDescriptionValidationService` : main service for DSL validation
  - `DSLValidationListener` : collects stats during tree walk
  - `DSLValidationVisitor` : visits grammar nodes and extracts version
  - `ValidationErrorListener` : ANTLR error collector
  - `ValidationResult` : encapsulates result, errors and parsed version

Validation Logic:

- Uses `DSL_GRAMMARLexer` + `DSL_GRAMMARParser`
- Walks tree with `ParseTreeWalker` and custom `DSLValidationListener`
- Extracts version with `DSLValidationVisitor`
- Applies additional validations:
  - At least 1 `figureDeclaration`
  - At least 1 action (`move`, `lightsOn`, etc.)

Show Description Generator Plugin (`show_high_level_description_plugin`)

- Combines validated figures into a complete DSL show.
- Input: `Proposta_mod_01.txt` + individual figure DSLs (`sample_DSL_figure_X.txt`)
- Output: `output/show_description.dsl`

Classes:

- `GenerateShowDslUI` : prompts for input file, parses figure files, builds proposal
- `DefaultShowGeneratorPlugin` :
  - Appends DSL version and figure DSLs to a `show {}` block
  - Handles file writing
- `ShowGeneratorPlugin` : interface for all show generators

Show Proposal Plugin (`ShowProposal_Plugin`)

- Validates the full proposal text file for drone shows.
- Based on ANTLR grammar: `ShowProposal.g4`

Classes:

- `ShowProposalValidationService`:
  - Validates either from string or file
  - Uses both listener and visitor
- `ShowProposalValidationListener`:
  - Ensures non-empty title and drone list
- `ShowProposalValidationVisitor`:
  - Ensures valid VAT and business rules

How to Build

```
# From each plugin's root directory
mvn clean install
```

How to Run

```
# Generate show DSL from a proposal
java show_high_level_description_plugin.GenerateShowDslUI

# Launch UI for validating drone programs (Language 2)
java lapr4.app.backoffice.console.presentation.dronemodel.DroneValidator_L2_UI

# Validate Language 1 program via UI
java lapr4.app.backoffice.console.presentation.dronemodel.DroneValidator_L1_UI
```

User Story Mapping

ID	Title	Covered	Component(s)
US340	DSL Plugin for figure description	Yes	DSL-PLUGIN, DroneScript (L2)
US341	Validate figure description	Yes	DSL-PLUGIN
US345	Drone language plugin per language	Yes	DroneScript L1 & L2
US347	Proposal validation and generation	Yes	ShowProposal_Plugin
US348	Generate show high-level description	Yes	show_high_level_description_plugin

---

Execution Flow

1. CRM or Tech writes a `Proposta_mod_01.txt` file with a list of figure names.
2. `GenerateShowDslUI` reads that file and extracts figure names.
3. Each figure DSL is loaded from `sample_DSL_figure_X.txt` and validated.
4. A `ShowProposal` is built with valid `FigureInShowProposal` objects.
5. `DefaultShowGeneratorPlugin` combines them into a full DSL in `show { ... }`.
6. Final DSL is saved to `output/show_description.dsl`.
7. Individual drone scripts may then be validated using Language 1 or Language 2 plugins.

Integration Points

- **Menu Entries**
  - Validate Drone Program (Language 1)
  - Validate Drone Program (Language 2)
  - Generate the Show high-level Description

Future Improvements

- Add semantic validation (e.g. drone count, figure duration).
- Display AST from ANTLR visually (for debugging/teaching).
- Improve error reporting (timestamps, severity levels).
- Support plugin discovery dynamically (using classpath scanning).

Authors & Contributors

- 1230596
- 1230595
- 1220848
- 1240588