

Sprint 3 LPROG

Turma : 2DJ

Grupo : 50

Jorge Sousa - 1210820

Rui Silva - 1221257

André Pereira – 1221275

Daniil Pogorielov – 1221647

Pedro Almeida – 1231499

US 340 - “DSL Plugin”

In this user story, we will implement a Domain-Specific Language (DSL) plugin that allows the system to interpret and process specific commands related to drone models. The plugin will be designed to facilitate the creation and manipulation of drone figures using a high-level language tailored for this purpose. The plugin will be integrated into the existing architecture, enabling seamless interaction with the drone model controller and ensuring that the DSL can be utilized effectively throughout the application.

US 341 - “Validate figure description”

In this user story, we will develop a validation mechanism for figure descriptions using the DSL. The validation process will ensure that the figure descriptions conform to the expected syntax and semantics defined by the DSL grammar. This will involve creating a validation controller that interacts with the DSL parser to check for errors in the figure descriptions. If the validation is successful, the system will proceed with the figure creation; otherwise, it will return error messages detailing the issues found.

US 344 - “Generation of a drone program”

In this user story, we will implement a feature to generate a drone program based on the validated figure description. The drone program generator will take the figure description and convert it into executable code that can be run by the drones. This process will involve mapping the high-level commands from the figure description to the specific programming constructs required for the drones. The generated program will be stored for later execution during the drone shows.

US 345 - “Drone language Plugin”

In this user story, we will create a plugin that supports various programming languages used by different drone models. This plugin will allow the system to interpret and validate the code generated for

each drone model, ensuring compatibility with the specific programming language requirements. The plugin will be integrated with the drone model controller, enabling it to handle language-specific features and facilitate the generation of drone programs in the appropriate language.

US 346 - “Validation of a drone program”

In this user story, we will implement a validation process for the generated drone programs. The validation will ensure that the programs adhere to the syntax and operational constraints of the respective programming languages. This will involve using the language plugin to parse and check the generated code for errors. If the validation is successful, the program will be marked as ready for execution; if not, detailed error messages will be provided to assist in debugging.

US 347 - “Proposal generation”

In this US, we will generate the final document of a show proposal using a predefined template and validate it using the ANTLR grammar that had been made in the previous Sprint.

At first, we created a controller to process both these things. This controller will obtain the id of the show proposal we want to create the document for and the id of the template that the proposal document will follow. Then, through the generator plugin we will convert the placeholders from the template (within brackets) into real values from the show proposal that we have. Later, the document will be created using these values and will be stored in the folder **outputfiles**.

Next, we validate the proposal through the validator plugin, that will use the parser and lexer generated by the ANTLR grammar we had created in the previous Sprint (grammar was slightly edited compared to last Sprint) to make this validation and in case of a success returns a success message and saves the document in the show proposal for persistence. In case of failure, it will return a message indicating the

failure of the validation and also prints the errors that made the validation fail.

US 348 - “Show generation”

In this user story, we will implement the functionality to generate a show based on the available figures and drone models. The show generator will take the details of the selected figures and the corresponding drone models to create a cohesive show plan. This process will involve coordinating the timing and sequence of the figures to ensure a smooth performance. The generated show will be stored in the system for future reference and execution during the actual drone shows.