Product Require Document

4 January 2023- 6 January 2023

Overview

The goal of this project is to add a new F# binding for FABGen, allowing users of this binding generator tool to use F#. Currently, FABGen supports several bindings for different programming languages such as GO, LUA and Python but there is no binding for F#.

Opportunity Statement

We want to implement F# in FABGen to allow non-coding experts to have access to this software. C++ is a very specific language for non-coding experts people so implementing other languages such as Python, F# and Rust allows other experts to easily use FABGen. Adding F# to FABgen will benefit to the F# users who need a 3D engine. In addition, people looking for an alternative to SWIG when binding a C++ library to F# might find it useful. F# is known for being a relatively concise and easy to learn language, which can allow you to write code more quickly and with fewer errors. Also F# uses the .NET "int" data type, which is natively optimized for mathematical calculations and bit operations, which can make F# code runs faster than code written in other languages.

Audience

All users of FABGen will benefit from it , especially those who are F# experts.

What will happen

Today: FABGen can be only use with Python, GO and LUA

Tomorrow: All users could have the possibility to use FABGen with an F# code.

Sizing

We believe this could be achieved within 7 weeks.

With a team of 5 people.

Selected solution

Here is the link of the github of the project FABGen x ALGOSUP
https://github.com/harfang3d/algosup-binding-project

We've decided to separate the solution in 3 steps :

Create a mapping of elementary types

Implement a C API wrapping the c/C++ objects

Improve the integration with the target language

Constraints

- The F# binding must be compatible with the latest stable version of F#
- The binding must be integrated into FABGen and made available to users

Milestones

Work should start on week #1 and finished on week #7

- Being able to run FABGen to generate the existing binding layers to Cpython and Lua.
- Being able to run FABGen to generate the binding to Go.
- Run the unit tests.
- Examine how the Go binding works.
- map the types and implement the features required by the tests, starting with the "easy" ones.
- Ultimately improve the integration with the target language.